GUIDA ALLA SOLUZIONE DEI PROBLEMI
OPS 2022

INDICE DEL DOCUMENTO

0. INTRODUZIONE

Il programma OPS per il 2021-2022 si rivolge, come di consueto, a tre livelli di partecipazione:

- scuola primaria,
- scuola secondaria di primo grado,
- scuola secondaria di secondo grado (primo biennio).

Sono previste *gare a squadre* per tutti i livelli e *gare individuali* per gli ultimi due livelli.

Ogni *gara a squadre* consisterà di norma in 13 problemi; l'articolazione dei problemi sarà, *usualmente*, la seguente:

1. cinque problemi formulati in italiano e scelti, di volta in volta, tra l'insieme dei "Problemi ricorrenti" (si veda il successivo elenco);
2. sei problemi formulati in italiano e relativi a uno pseudo-linguaggio di programmazione;
3. un problema di comprensione di un testo in lingua italiana;
4. un problema formulato in inglese, di argomento ogni volta diverso (almeno in linea di principio).

Ogni *gara individuale* consisterà di 8 problemi; suddivisi come segue:

1. quattro problemi formulati in italiano e scelti, di volta in volta, tra l'insieme dei "Problemi ricorrenti" (si veda il successivo elenco);
2. tre problemi formulati in italiano e relativi a uno pseudo-linguaggio di programmazione;
3. un problema formulato in inglese, di argomento ogni volta diverso (almeno in linea di principio).

La difficoltà e la complessità dei problemi saranno commisurate al livello cui tali problemi saranno proposti.

I *Problemi ricorrenti* nelle gare OPS 2019-2020 sono tratti del seguente insieme:

a) Regole e deduzioni.
b) Fatti e conclusioni.
c) Grafi.
d) *Knapsack*.
e) Pianificazione.
f) Crittografia.
g) Movimenti di un robot.
h) Sottosequenze.
i) Flussi in una rete di canali.

Il seguito di questo documento si articola in quattro parti.

Il paragrafo 1 illustra brevemente come scrivere i numeri, i *termini* e le *liste* (questi ultimi sono "scritture formali" che possono essere utilizzate nei testi dei problemi proposti e nelle soluzioni richieste).

Il paragrafo 2 elenca le regole generali per compilare le risposte ai problemi: tali regole sono *stringenti*, perché la correzione dei problemi viene fatta in automatico.

Il paragrafo 3 riportata esempi dei problemi ricorrenti; tali esempi sono di norma due: uno orientato alla scuola primaria e uno orientato alla scuola secondaria (se è fornito un solo esempio il problema ricorrente è orientato a un solo ordine di scuola). I problemi sono di carattere abbastanza semplice; servono principalmente di riferimento per:

- fissare gli argomenti e la terminologia usata,
- tratteggiare i metodi di soluzione,
- mantenere "sintetico" l'enunciato dei problemi assegnati effettivamente in gara.

L'ultimo paragrafo illustra brevemente gli *elementi di pseudolinguaggio* che saranno utilizzati nei problemi assegnati in gara. La trattazione è divisa in due parti: i sottocapitoli 4.0 e 4.1 riguardano i tre livelli scolastici, mentre nei restanti gli argomenti trattati sono orientati alla scuola secondaria (non escludendo comunque un loro semplice utilizzo anche nella primaria).

N.B. Si suppone che i partecipanti alle gare abbiano letto il presente materiale e, comunque, l'abbiano a disposizione durante lo svolgimento della gara.

1. ALCUNE REGOLE DI SCRITTURA

Nei testi dei problemi possono comparire: *numeri*, *termini* e *liste*; le seguenti osservazioni sono utili per comprenderne il significato e l'uso corretto.

1.1 NUMERI

I numeri naturali maggiori di nove sono scritti utilizzando le dieci cifre indo-arabe 0,1,2,3,4,5,6,7,8,9 e sono letti da sinistra verso destra.

Unica eccezione alla regola è che un naturale non può iniziare con lo "0".

Talvolta per facilitare la lettura di numeri molto lunghi (con più di 5 cifre), si scrivono utilizzando un *separatore periodico* ogni tre cifre a partire da destra.

Nella *lingua italiana* si utilizza come separatore il punto nella parte inferiore della riga:

12.345

1.234.556.789

In lingua inglese il separatore è la virgola:

12,345

1,234,556,789

N.B. Nella compilazione delle risposte i numeri interi sono *sempre* scritti *senza* separatore periodico e *senza* zeri a sinistra.

Esempio. In un certo esercizio la casella di risposta permette di scrivere numeri naturali fino a cinque cifre, mentre la risposta al quesito è 123.

Risposta corretta se si scrive 123 (lasciando vuoti i due spazi a destra del 3)

Risposta sbagliata se si scrive 0123 oppure 00123 (volendo completare i cinque spazi a disposizione)

I NUMERI RAZIONALI nella forma decimale sono sequenze di cifre separate da un simbolo detto *marcatore.* La parte a sinistra del marcatore è detta *parte intera*, mentre quella a destra è detta *parte decimale.*

Nella *lingua italiana* il marcatore è la *virgola* nella parte bassa del rigo:

0,23 parte intera 0; parte decimale 23

2,343 parte intera 2; parte decimale 343

45,0 parte intera 45; parte decimale 0

Nella *lingua inglese* il marcatore è il *punto* nella parte bassa del rigo:

0.23

2.343

45.0

N.B. Nella compilazione delle risposte il numero *intero* 45 è "diverso" dal numero *decimale* 45,0 (o 45.0 se l'ambito è in lingua inglese): il testo del problema specificherà sempre la natura della risposta richiesta.

I numeri razionali sono rappresentabili anche in forma di frazione (due numeri interi separati dal simbolo di frazione /. Il numero a sinistra è detto numeratore e quello a destra denominatore)

Esempio. 5/6 è una frazione con numeratore 5 e denominatore 6

21/5 è una frazione con numeratore 21 e denominatore 5

Il passaggio dalla forma frazionaria alla forma decimale avviene effettuando una divisione dove il dividendo è il numeratore e il divisore è il denominatore.

Esempio.

21/5 corrisponde al decimale 4,2 (21:5 = 4,2)

2/3 corrisponde al decimale 0,666666……. (2:3=0,666666…. divisione che non si arresta)

3/15625 corrisponde al decimale 0,000192 (3:15625=0,000192)

In genere un numero razionale viene approssimato scrivendo le sue prime cifre decimali. I metodi utilizzati sono due: troncamento e arrotondamento

Per *troncamento* di un numero razionale ad una certa cifra decimale si intende il numero scritto nella forma decimale fino a quella cifra.

Esempio. Scrivere i seguenti numeri razionali troncati alla seconda cifra decimale

Numero razionale	Numero troncato alla seconda cifra decimale
9,3435	9,34
9,3402	9,34
8,9951	8,99
0,898	0,89
0,9	0,90

Per *arrotondamento* di un numero razionale ad una certa cifra decimale si intende il numero scritto nella forma decimale utilizzando la seguente regola:

se la cifra successiva (a destra della cifra decimale stabilita) è 0,1,2,3,4 allora si scrive il numero troncato

se la cifra successiva è 5,6,7,8,9 allora si aggiunge 1 (alla cifra decimale stabilita) e si scrive il nuovo numero.

Esempio1. Scrivere il numero decimale 9,3435 arrotondato alla seconda cifra decimale
R. 9,34 (perché la terza cifra decimale è 3 e quindi si tronca semplicemente)

Esempio2. Scrivere il numero decimale 9,3435 arrotondato alla terza cifra decimale
R. 9,344 (perché la quarta cifra è 5 per cui la terza cifra 3 viene aumentata di 1)

Esempio3. Scrivere il numero decimale 7,9953 arrotondato alla seconda cifra decimale
R. 8,00 (perché la terza cifra è 5 e aumentando di 1 la seconda cifra si hanno dei riporti)

Nelle ultime edizioni OPS, in problemi coinvolgenti soluzioni con numeri decimali, c'è la tendenza a chiedere come risposta un numero intero.
Precisamente si chiede di *"arrotondare il numero razionale all'intero più vicino"*

In questo caso si esegue la regola dell'arrotondamento vista sopra, assumendo come cifra successiva la prima cifra decimale.
Alcuni esempi
1) Arrotondare all'intero più vicino il numero 67,26
R. 67 (perché la prima cifra decimale è 2)

2) Arrotondare all'intero più vicino il numero 7,63
R. 8 (perché la prima cifra decimale è 6 e pertanto si aggiunge 1 alla parte intera che è 7)

Per completezza ricordiamo che le stesse regole si applicano anche ai numeri negativi
-67,2 viene arrotondato all'intero -67 (perché la cifra decimale è 2)
-54,9 viene arrotondato all'intero -55 (perché la cifra decimale è 9)

N.B.1 La risposta su una percentuale (%) viene sempre data come arrotondamento all'intero più vicino
Esempio. Esprimere in % la frazione 23/79
R. 29% (23:79 = 0,291; 0,291*100 =29,1 e si arrotonda all'intero più vicino 29)

N.B.2 La regola dell'arrotondamento all'intero più vicino si applica anche ai problemi in lingua inglese dove la frase
Write your answer as an integer number (eventually rounded up to the nearest whole integer)
viene interpretata come "scrivere in risposta l'intero più vicino al numero decimale trovato"

1.2 DATE ED ORE

Il testo del problema e/o le diciture che accompagnano i campi da riempire, specificheranno in maniera puntuale come indicare date e ore.

1.3 I TERMINI

Un *termine* è una scrittura del tipo:

minerale(m1,90,300)
attività(alfa,4,6)

cioè un *nome* seguito da *uno* o più *argomenti* racchiusi tra parentesi tonde e separati da virgole (se sono più di uno).
Il *nome* inizia sempre con una lettera; gli *argomenti* sono parole, sigle o numeri interi
N.B. Nella scrittura di un termine non intervengono spazi tra nome e parentesi, tra argomento e virgola, tra argomento e parentesi. Dunque le seguenti scritture sono sbagliate

minerale (m1,90,300) c'è spazio tra nome e parentesi
minerale(m1, 90,300) c'è spazio tra il secondo argomento e la prima virgola
minerale(m1,90,300) c'è spazio tra il terzo argomento e l'ultima parentesi tonda

Prima dell'uso, ogni termine è definito. La definizione indica il *nome* del termine, gli *argomenti* e il loro *significato*.

La tabella seguente che si riferisce ad un magazzino di minerali: riporta la sigla, il peso in kg e il valore in euro di ogni esemplare presente nel magazzino.

minerale		
sigla	**valore in euro**	**peso in kg**
m1	213	30
m2	200	35
m3	230	38

Il nome del termine è il nome della tabella, il numero degli argomenti è il numero delle colonne e il significato di ogni termine è l'intestazione della colonna.

Lo stesso magazzino può essere descritto attraverso la seguente definizione:

minerale(<sigla>,<valore in euro>,<peso in kg>)

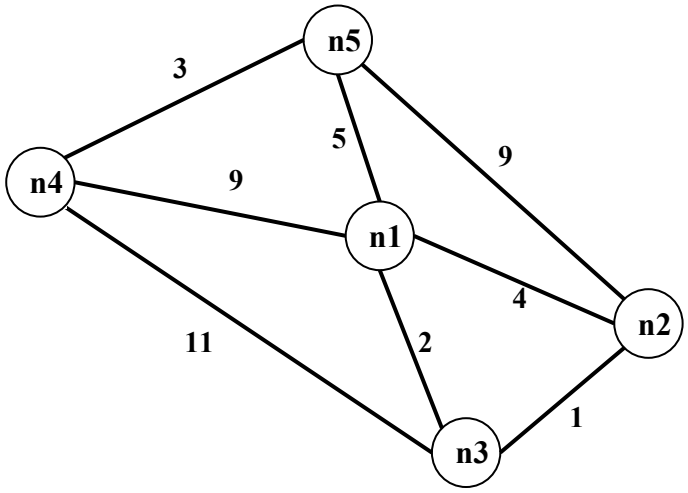
che mostra il *nome* del termine, il *numero* degli *argomenti* e il loro *significato*. Di conseguenza il contenuto della tabella viene così descritto:

minerale(m1,213,30)
minerale(m2,200,35)
minerale(m3,230,38)

Come ulteriore esempio si consideri la seguente figura che rappresenta un *grafo*; questo è composto da *nodi*, distinti da una sigla e da *archi* che congiungono i nodi.
Un grafo si può pensare come la rappresentazione schematica di città congiunte da strade.
Un grafo può essere descritto con dei termini; ogni termine è associato a un arco ed è così *definito*:

arco(<nodo1>,<nodo2>,<lunghezza>)

La definizione mostra il nome del termine e il numero degli argomenti (tre): i primi due si riferiscono ai nodi posti agli estremi dell'arco, il terzo descrive la lunghezza dell'arco.



I termini che descrivono il grafo in figura sono:

arco(n1,n2,4)	arco(n1,n3,2)	arco(n1,n4,9)	arco(n1,n5,5)
arco(n4,n5,3)	arco(n4,n3,11)	arco(n2,n3,1)	arco(n2,n5,9)

1.4 LE LISTE

Una *lista* è una scrittura del tipo:

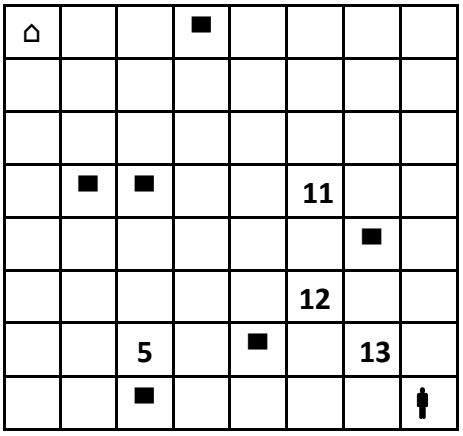
[alfa,beta,gamma]
[a1,2,56,b2,3,67]

cioè una coppia di parentesi quadre che racchiudono un certo numero di *elementi* separati da virgole (se sono più di uno). Una lista può non contenere elementi: in tal caso si dice *lista vuota* [].

N.B. Nella scrittura di una lista non intervengono spazi.

Gli *elementi* sono parole, sigle, numeri interi o *altre liste*.

Si consideri la seguente figura che rappresenta una scacchiera 8x8. Facendo riferimento allo spigolo in basso a sinistra, ogni casella può essere individuata da una coppia di coordinate, scritte come una lista, il cui primo elemento è la "X", cioè lo spostamento orizzontale e il secondo elemento è la "Y", cioè lo spostamento verticale.
Per esempio il numero 5 è nella casella [3,2]; il robot è nella casella [8,1] e la casetta è nella casella [1,8].



Per indicare la posizione sulla scacchiera dei quadrati neri si può usare una lista i cui elementi sono le coordinate delle caselle che li contengono; si ottiene, quindi, una lista di liste:

[[2,5],[3,1],[3,5],[4,8],[5,2],[7,4]]

Volendo indicare la posizione sulla scacchiera e il valore dei quattro numeri si può usare una lista di quattro elementi che, a loro volta, sono liste; ciascuna lista "interna" è formata dalle coordinate della casella che contiene il numero seguite dal valore:

[[3,2,5],[6,3,12],[6,5,11],[7,2,13]]

Per indicare la posizione della freccia →
sulla scacchiera si può usare la lista vuota:

[]

per significare, appunto, che non compare.

2. LA COMPILAZIONE DELLE RISPOSTE

Con il termine *stringa* intendiamo una qualunque sequenza di simboli anche ripetuti.
La correzione dei problemi viene fatta da un sistema automatico confrontando la stringa, proposta come risposta ad un quesito, con la stringa che rappresenta la risposta attesa (o "esatta").
È quindi importante la compilazione, rispettando *esattamente* le indicazioni riportate nel testo dei problemi; inoltre, si devono sempre osservare le seguenti regole generali.

1. In ciascun campo disponibile per la risposta è possibile, *di norma*, digitare, allineato a sinistra, un solo elemento: parola, numero, lista o termine; le parole devono essere scritte *senza accenti*; per i numeri si veda il paragrafo precedente.
2. Talvolta il testo del problema può specificare che la risposta è una frase: in tal caso occorre scriverla con *un solo spazio* tra due parole consecutive.
3. Gli elementi che compongono una lista vanno riportati fra parentesi quadre separati da virgole *senza spazi*: per esempio la lista delle prime tre lettere (minuscole) dell'alfabeto va scritta nel modo seguente: [a,b,c] e non [a, b, c]; i campi, per le risposte che richiedono liste, riportano già le parentesi quadre "esterne"; se gli elementi delle lista sono, a loro volta, liste occorre scrivere le opportune parentesi che li racchiudono.
4. L'elevazione a potenza è indicata da x^n; per esempio: x^2 deve essere scritto x^2.
5. Una frazione deve essere scritta su una linea; per esempio: 2**/**3.
6. Se nel testo del problema è indicata una particolare modalità di risposta, per esempio:

digitare V per vero o F per falso

7. è necessario attenersi strettamente alle istruzioni: in caso contrario il sistema registrerà una risposta errata.
8. Gli allenamenti disponibili sul sito consentono di familiarizzarsi con le regole di compilazione delle risposte.

3. PROBLEMI RICORRENTI

a) REGOLE E DEDUZIONI

PREMESSA
Per risolvere problemi spesso esistono delle regole che, dai dati del problema, permettono di calcolare o *dedurre* la soluzione.
Questa situazione si può descrivere col termine

regola(<sigla>,<lista degli antecedenti>,<conseguente>)

che indica una regola di nome <sigla> che consente di dedurre <conseguente> conoscendo tutti gli elementi contenuti nella <lista degli antecedenti>, detta anche *premessa*. Problemi "facili" possono essere risolti con una sola regola; per problemi "difficili" una sola regola non basta a risolverli, ma occorre applicarne diverse in successione.

Consideriamo le seguenti regole (a rigore le regole associate ai seguenti termini):

regola(1,[e,f],b)	regola(2,[m,f],e)	regola(3,[m],f)
regola(4,[b,f],g)	regola(5,[b,g],c)	regola(6,[g,f],c)

La regola 1 dice che si può calcolare (o dedurre) **b** conoscendo **e** ed **f** (cioè gli elementi della lista [e,f]); conoscendo **b** ed **f** (cioè gli elementi della lista [b,f]) è possibile dedurre **g** con la regola 4. Quindi, a partire da **e** ed **f** è possibile dedurre prima **b** (con la regola 1) e poi **g** (con la regola 4).
N.B. I due seguenti termini:

regola(1,[e,f],b) regola(1,[f,e],b)

individuano la stessa regola (calcolare b conoscendo e ed f).

Un *procedimento di deduzione* (o deduttivo, o di calcolo) è rappresentato da un *insieme di regole da applicare in sequenza opportuna* per dedurre un certo elemento (incognito) a partire da certi dati: quindi può essere descritto dalla lista delle sigle di queste regole. Il procedimento [1,4] descrive la soluzione del problema: "dedurre **g** a partire da **e** ed **f**".
Una maniera grafica per rappresentare le regole è quella mostrata nella seguente figura 1: consiste in un simbolo (da leggere dal basso verso l'alto) costituito da:

un cerchio con all'interno scritto il numero della regola
in basso frecce (con una sigla all'inizio), la cui punta tocca il cerchio; ogni freccia è un antecedente
in alto dal cerchio "esce una freccia" sulla cui punta vi è scritto il conseguente.

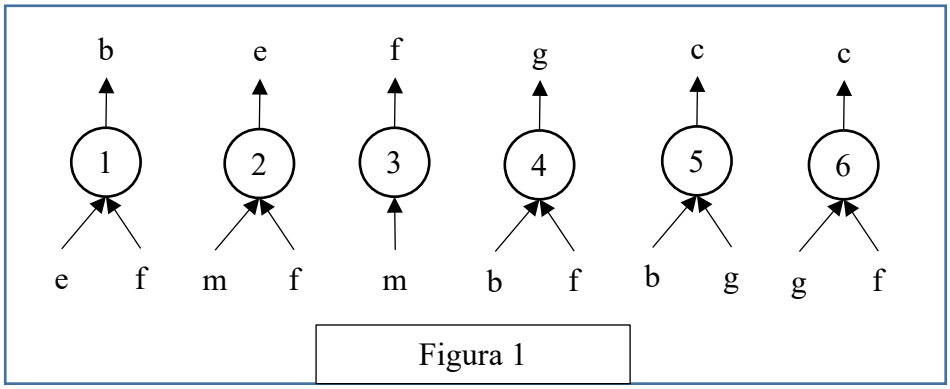


Figura 1

Con questa rappresentazione grafica, risolvere il problema "dedurre **g** a partire da **e** ed **f** "
è particolarmente facile; si cerca una regola che ha per conseguente **g**.
In questo caso esiste solo la regola 4 (figura 2 a sinistra).

Dei due antecedenti **b** ed **f** solo il secondo è noto: quello incognito (**b** in questo caso) va considerato come conseguente di una nuova regola. Nel nostro caso quindi bisogna utilizzare la regola 1 in cui il conseguente **b**, si deduce a partire dagli antecedenti **e** ed **f.**
Il procedimento è quindi (individuato dalla lista) [1,4].

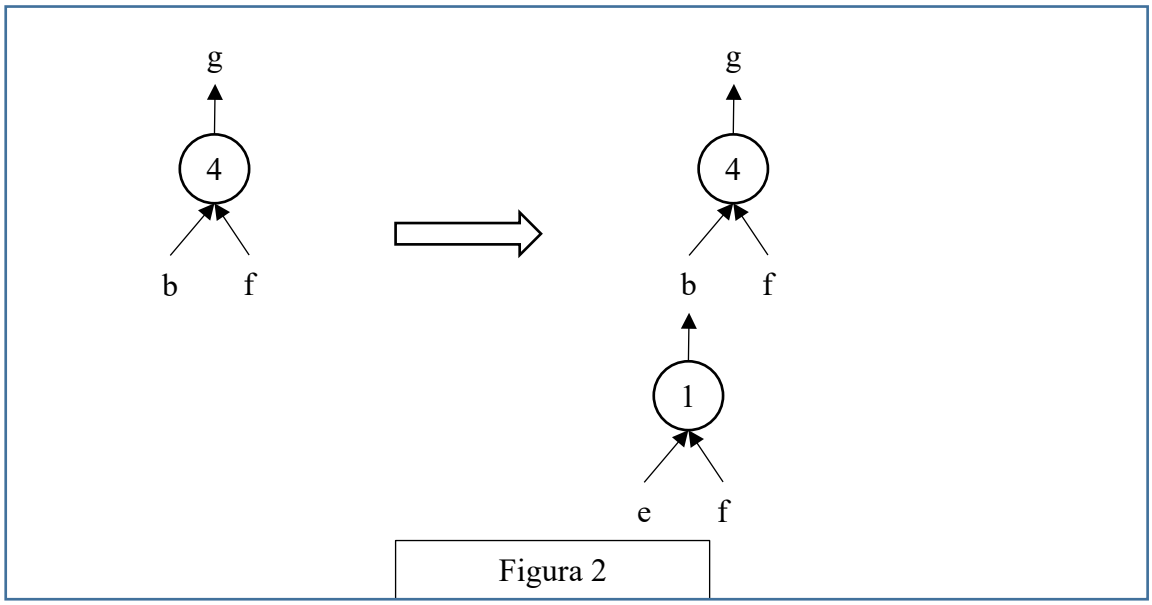


Figura 2

N.B. Nelle liste richieste occorre elencare le sigle delle regole nell'ordine che corrisponde alla sequenza di applicazione: la prima (a sinistra) della lista deve essere la sigla che corrisponde alla prima regola da applicare (che ha come antecedenti solo dati); l'ultima (a destra) deve essere la sigla della regola che ha come conseguente l'elemento incognito da dedurre.
Nella lista non ci sono regole *ripetute* (infatti un procedimento di deduzione è un *insieme* di regole da applicare in opportuna sequenza). L'applicazione di una regola rende disponibile il conseguente da utilizzare (come antecedente) nell'applicazione di regole successive.
La lista associata a un (ben preciso) procedimento si costruisce quindi per passi successivi a partire dal primo elemento che è la sigla della prima regola da applicare.
Se ad un certo passo, ci sono più regole applicabili, occorre dare la precedenza (nella lista) a quella con sigla *inferiore* (questo per rendere *unica* la lista associata al procedimento).

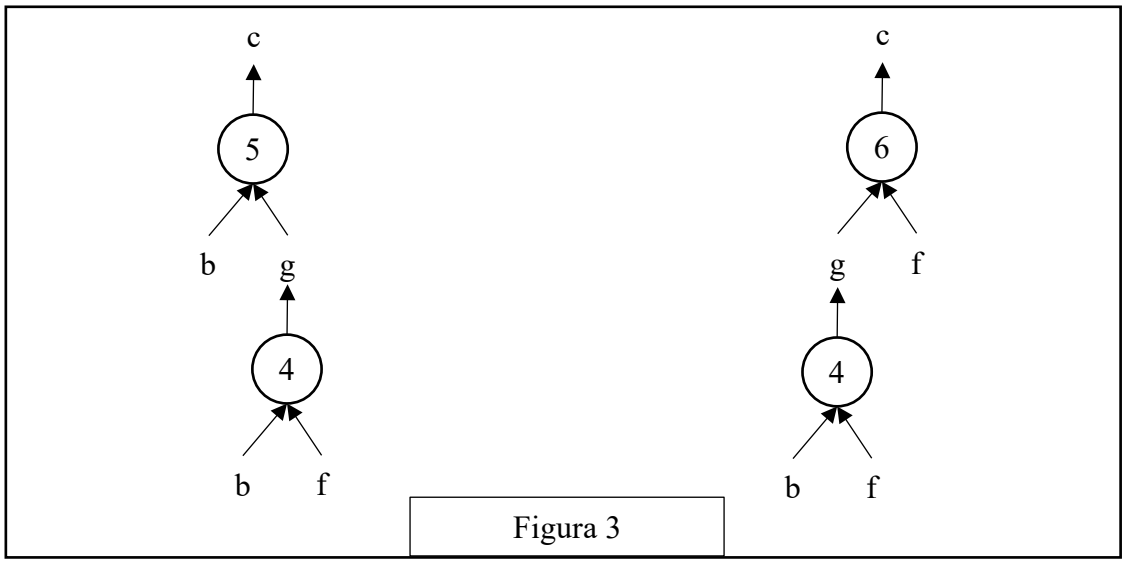


Figura 3

N.B. In alcuni casi esistono più procedimenti deduttivi possibili che permettono di ricavare un certo elemento dagli stessi dati, in maniere diverse (cioè con alberi diversi e quindi con insiemi diversi di regole). Per esempio il problema "dedurre **c** a partire da **b** ed **f**" (dalle regole viste sopra) ha due distinti procedimenti risolutivi; gli alberi relativi ai due procedimenti sono mostrati nella seguente figura 3. Le liste associate sono, rispettivamente, [4,5] e [4,6].

In un procedimento deduttivo, il numero di regole *differenti* coinvolte (e, quindi, anche il numero di elementi della lista corrispondente al procedimento) si dice *lunghezza* del procedimento.

ESEMPIO 1
Siano date le seguenti regole:

regola(1,[b,c],a) regola(2,[c,d],a)
regola(3,[b,c,d],a) regola(4,[b,a],f)

Trovare:
1. la sigla N della regola che consente di dedurre **a** da **d** e **c**;
2. la lista L che rappresenta il procedimento per dedurre **f** da **b** e **c**.

SOLUZIONE

N	2
L	[1,4]

COMMENTI ALLA SOLUZIONE
Per rispondere alle due semplici domande è opportuno applicare il metodo *backward*, cioè è opportuno partire dalla incognita (l'elemento che occorre dedurre) e cercarlo nel conseguente delle varie regole.
Per la prima domanda (che chiede di dedurre **a**) si osservi che le regole 1, 2 e 3 hanno tutte come conseguente **a**, e quindi permettono di dedurlo; ma solo la regola 2 ha come premessa [c,d] cioè ha come antecedenti i dati (**d** e **c**) della prima domanda: quindi è quella la regola cercata.
Per rispondere alla seconda domanda, si osservi come una sola regola, la 4, ha come conseguente **f**; ma ha come antecedenti **b** e **a**; però, di questi, solo **b** è noto: quindi, per poter applicare tale regola, occorre dedurre **a**; come già osservato, le regole 1, 2 e 3 hanno tutte come conseguente **a**, e quindi permettono di dedurlo; ma adesso (a differenza della prima domanda) sono noti **b** e **c**: quindi stavolta è la regola 1 quella che deve essere applicata. Nel costruire la lista richiesta si ricordi che il primo elemento di tale lista è la prima regola che deve essere applicata, quindi (tutti) i suoi antecedenti devono essere dati.

ESEMPIO 2
Siano date le seguenti regole:

regola(1,[u,d],c) regola(2,[q,n],g) regola(3,[p,q],n)
regola(4,[c,d],z) regola(5,[u],d) regola(6,[n,u],m)

Trovare:
1. la lista L1 che descrive il procedimento per dedurre **g** a partire da **p** e **q**;
2. la lista L2 che descrive il procedimento per dedurre **z** a partire da **u.**

.

SOLUZIONE

L1	[3,2]
L2	[5,1,4]

COMMENTI ALLA SOLUZIONE

Per risolvere questo tipo di problemi si può usare il metodo *backward* (o *top down*) che consiste nel partire dalla incognita e cercare di individuare una regola per derivarla. Se esiste una regola i cui antecedenti sono tutti noti (i dati) la soluzione è trovata; altrimenti si cerca una regola i cui antecedenti non sono tutti noti e si continua a cercare regole per derivare gli antecedenti incogniti (che compaiono nella premessa).

Per la prima domanda si verifica immediatamente che **g** compare come conseguente nella regola 2 che ha come antecedenti **q** (dato) e **n** (incognito). Occorre quindi dedurre **n**: questo è conseguente solo della regola 3, che ha come antecedenti **p** e **q** entrambi dati. Quindi la lista L1 è [3,2]; si noti l'ordine delle regole: la prima che compare (a sinistra) nella lista è la prima da applicare e l'ultima trovata col metodo *backward*.
Per la seconda domanda, di nuovo si può osservare che **z** compare come conseguente nella regola 4 che ha come antecedenti **c** e **d**: entrambi incogniti. È facile vedere che **d** può essere dedotto con la regola 5 da **u** (noto); poi noto anche **d**, con la regola 1 si deduce **c**. Quindi la lista L2 è [5,1,4].

N.B. La prima regola che compare nella lista (che rappresenta il procedimento) ha come antecedenti solo dati; la seconda e le successive hanno antecedenti presi dai dati o dagli elementi dedotti mediante le regole che compaiono precedentemente nella lista. L'ultima regola ha come conseguente l'elemento cercato.

ESEMPIO 3

La scrittura *mix(1,[azzurro,giallo],verde)* indica che la combinazione 1 può essere applicata per ottenere il risultato "verde" partendo dai due colori "azzurro" e "giallo". Un procedimento deduttivo permette di concatenare fra loro combinazioni di cui siano noti i dati di partenza per giungere alla soluzione di un problema.

Consideriamo le seguenti combinazioni che producono colori:

mix(1,[azzurro,giallo],verde)
mix(2,[blu, rosso],viola)
mix(3,[verde,blu],turchese)

Trovare:

1. la sigla N della regola che consente di ricavare il viola dal blu e dal rosso;
2. la lista L che rappresenta il procedimento per ricavare il turchese partendo da: azzurro, giallo e blu.

SOLUZIONE

N	2
L	[1,3]

COMMENTI ALLA SOLUZIONE

Per rispondere alle due domande è opportuno partire dall'incognita (l'elemento che occorre dedurre) e cercarlo nel conseguente delle varie regole.
Per la prima domanda (che chiede di ricavare il viola) si osservi che proprio la regola 2 ha come colori di partenza blu e rosso e come incognita il colore richiesto viola: quindi è questa la regola cercata.
Per rispondere alla seconda domanda, si osservi che è possibile concatenare la regola 1 (che permette di ottenere il verde partendo dai colori che abbiamo azzurro e giallo) e la regola 3 (che permette di ottenere il

turchese partendo dal colore verde, appena ottenuto, e dal colore blu che abbiamo) ottenendo appunto il colore turchese richiesto.

b) FATTI E CONCLUSIONI

PREMESSA

Questi problemi trattano di *entità* correlate da fatti; ciascuna entità ha *valori* discreti. Consideriamo, per esempio, le entità "nome", "cognome", "età"; se si parla di tre persone, allora il nome può avere i valori: Aldo, Giacomo, Giovanni; il cognome può avere i valori Storti, Poretti, Baglio e l'età i valori: 58, 59, 60. Nei problemi vengono enunciati dei fatti e da questi occorre *ragionare* e trarre *conclusioni* per associare opportunamente i valori di nome, cognome ed età.

Per risolvere questi problemi è utile tracciare una tabella.

La tabella si completa esaminando ognuno dei fatti e traendone le conseguenze utilizzando essenzialmente il principio del sillogismo ternario: se $x \in A \cap B$ e $x \in B \cap C$ allora $x \in A \cap C$.

PROBLEMA 1

Alice, Bastiano e Carla abitano nella stessa strada. I loro cognomi sono Rossi, Verdi e Bianchi, e le loro età sono 17, 19 e 20. I nomi, i cognomi e le età sono elencati in ordine casuale (e quindi non si corrispondono ordinatamente).

Dai due fatti elencati di seguito, determinare il nome completo e l'età di ogni persona, riempiendo la successiva tabella.

1. La signorina Rossi è tre anni più vecchia di Carla.
2. La persona di cognome Bianchi ha 19 anni.

NOMI	COGNOMI	ETÀ
Alice		
Bastiano		
Carla		

SOLUZIONE

NOMI	COGNOMI	ETÀ
Alice	Rossi	20
Bastiano	Bianchi	19
Carla	Verdi	17

COMMENTI ALLA SOLUZIONE

Fatto 1 : Rossi è una signorina e ha 20 anni mentre Carla ne ha 17.
Rossi è Alice perché la terza persona (Bastiano) è maschio.

NOMI	COGNOMI	ETÀ
Alice	Rossi	20
Bastiano		
Carla		17

Fatto 2 : Bianchi ha 19 anni. Allora Bianchi è Bastiano e Carla ha cognome Verdi.
Questo permette di completare la tabella.

NOMI	COGNOMI	ETÀ
Alice	Rossi	20
Bastiano	Bianchi	19
Carla	Verdi	17

PROBLEMA 2

Alice, Bastiano e Carla sono pronti per i Grandi Giochi Estivi. Vivono in tre grandi città: Roma, Napoli e Milano; si sono allenati duramente: chi per 3 mesi, chi per 5 mesi chi, addirittura per 7 mesi; praticano il golf, il canottaggio e il ciclismo.

Dai fatti elencati di seguito, determinare dove vive ogni atleta, per quanto tempo si è allenato e che sport pratica, riempiendo la successiva tabella.

1. Chi pratica golf si è allenato per 3 mesi.
2. Bastiano, non è quello che si è allenato per 5 mesi, pratica il ciclismo.
3. Alice si è allenata per due mesi più dell'atleta che sta a Roma.
4. Chi pratica canottaggio non sta a Milano.

NOMI	CITTÀ	SPORT	TEMPO
Alice			
Bastiano			
Carla			

SOLUZIONE

NOMI	CITTÀ	SPORT	TEMPO
Alice	Napoli	canottaggio	5
Bastiano	Milano	ciclismo	7
Carla	Roma	golf	3

COMMENTI ALLA SOLUZIONE

Fatto 1 : Sport : golf -------> Tempo 3 mesi

Fatto 2 : Bastiano pratica il ciclismo e si è allenato per 7 mesi

NOMI	CITTÀ	SPORT	TEMPO
Alice			
Bastiano		ciclismo	7
Carla			

Fatto 3 : Alice non abita a Roma e si è allenata per 5 mesi

NOMI	CITTÀ	SPORT	TEMPO
Alice			5
Bastiano		ciclismo	7
Carla			

Di conseguenza Carla si è allenata per 3 mesi e dal Fatto1 pratica golf e abita a Roma.

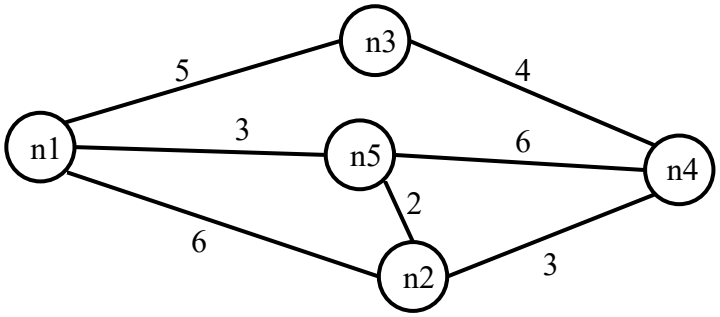
NOMI	CITTÀ	SPORT	TEMPO
Alice			5
Bastiano		ciclismo	7
Carla	Roma	golf	3

Fatto 4: Sport : canottaggio ------> Città : Napoli. Sono fatti che riguardano Alice.
Pertanto Bastiano abita a Milano.
Questo completa la tabella.

NOMI	CITTÀ	SPORT	TEMPO
Alice	Napoli	canottaggio	5
Bastiano	Milano	ciclismo	7
Carla	Roma	golf	3

c) GRAFI

Un *grafo* si può pensare come l'astrazione di una carta geografica: per esempio il grafo rappresentato nella figura seguente, descrive i collegamenti esistenti fra alcune (5) città: queste sono rappresentate da *nodi* di nome n1, n2, …, n5 e i collegamenti sono rappresentati da segmenti tra i nodi, detti *archi*.



A seconda del problema, gli archi possono essere percorsi in entrambe le direzioni (e in questo caso si parla di archi non-diretti) oppure solo in una (archi diretti). Gli archi diretti si rappresentano mediante una freccia, che va dal nodo di partenza a quello di destinazione.
In alcuni problemi, a ogni arco è associata una lunghezza, ovvero un numero, detta anche *peso* dell'arco. Quando gli archi di un grafo hanno un peso, si dice che sono *pesati* e i pesi degli archi vengono rappresentati come numeri scritti vicino alle frecce.
Il grafo rappresentato in figura ha archi non-diretti e pesati.
Come abbiamo visto nel paragrafo 1.3, un grafo può essere descritto mediante un elenco di termini, ciascuno dei quali definisce un arco tra due nodi del grafo. Nel caso di grafi con archi non pesati, si usano termini con due argomenti. I due argomenti sono i nomi dei nodi connessi dall'arco. Spesso (ma non in tutti i problemi!) si userà il termine "arco" per archi non diretti e "freccia" per archi diretti. Quindi un arco non diretto e non pesato, che connette i nodi **x** ed **y**, sarà descritto dal termine **arco(x,y),** mentre un arco diretto e non pesato, che connette i nodi **Bologna** e **Roma** sarà descritto dal termine **freccia(Bologna,Roma)**.
Nel caso di grafi con archi pesati, è necessario descrivere il peso, oltre che i due nodi collegati. Per questo motivo si useranno termini con 3 argomenti: i primi due sono i nomi dei nodi collegati e il terzo è un numero che rappresenta il valore del peso.
Il grafo rappresentato dalla precedente figura, che ha archi non diretti e pesati, viene quindi descritto dal seguente insieme di termini:

arco(n1,n2,6) **arco(n1,n3,5)** **arco(n3,n4,4)**
arco(n1,n5,3) **arco(n2,n4,3)** **arco(n2,n5,2)**
arco(n5,n4,6)

Due nodi si dicono *adiacenti* se sono collegati da un arco.
Il numero di archi che "escono" da un nodo si dice *grado* del nodo; per esempio nel grafo in figura, il nodo n3 ha grado 2, gli altri hanno grado 3.
Un *percorso* (o *cammino*) tra due nodi del grafo consiste in una sequenza di nodi ciascuno dei quali (tranne l'ultimo) è adiacente con il successivo; un percorso può, quindi essere descritto con una lista di nodi (quelli toccati dal percorso, ordinata dal nodo di partenza al nodo di arrivo). Per esempio, la lista [n5,n2,n4,n3] descrive un percorso dal nodo n5 al nodo n3; tale percorso ha lunghezza K = 2 + 3 + 4 = 9.
Un *ciclo* è un percorso che inizia e termina nello stesso nodo, per esempio [n5,n2,n1,n5].
Un percorso si dice *semplice* se *non* ha nodi ripetuti: un percorso semplice, quindi, non contiene cicli; per esempio [n5,n2,n4,n3] è semplice, mentre [n5,n2,n1,n5,n2,n4,n3] non è semplice perché ha nodi ripetuti.

N.B. Dato un grafo, come quello della precedente figura, è facile scrivere l'insieme di termini che lo descrivono; viceversa, disegnare il grafo a partire dall'elenco dei termini è meno ovvio (si vedano i problemi seguenti).

ESEMPIO 1
È dato un grafo con archi non diretti e pesati, descritto dal seguente elenco di termini:

arco(n1,n2,2) arco(n2,n3,2) arco(n3,n1,5)
arco(n4,n1,1) arco(n4,n2,4) arco(n4,n5,3)

Disegnare il grafo e trovare:

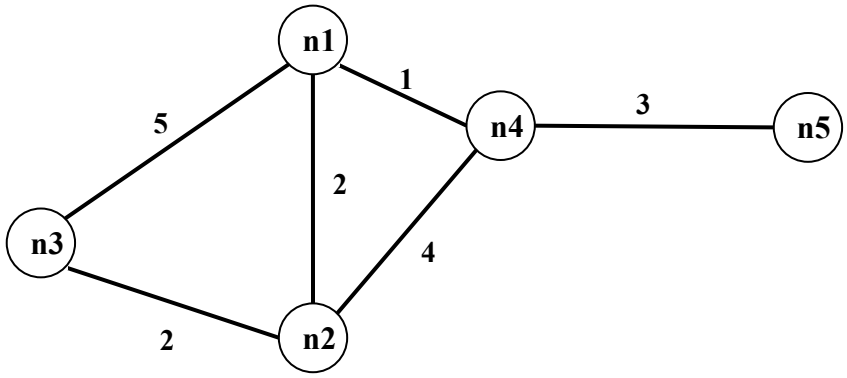
1. la lista L1 del percorso semplice più breve tra n5 e n3 e calcolarne la lunghezza K1;
2. la lista L2 del percorso semplice più lungo tra n5 e n3 e calcolarne la lunghezza K2.

L1	
K1	
L2	
K2	

SOLUZIONE

L1	[n5,n4,n1,n2,n3]
K1	8
L2	[n5,n4,n2,n1,n3]
K2	14

COMMENTI ALLA SOLUZIONE
Per disegnare il grafo si osservi innanzitutto che vengono menzionati 5 nodi (n1, n2, n3, n4, n5); si procede per tentativi: si disegnano i 5 punti nel piano e li si collega con archi costituiti da segmenti: probabilmente al primo tentativo gli archi si incrociano; si cerca poi di risistemare i punti in modo da evitare gli incroci degli archi: spesso questo si può fare in più modi. Da ultimo si riportano le distanze sugli archi, come mostrato dalla figura seguente.



Si noti che le lunghezze degli archi che compaiono nei termini (che rappresentano delle strade) *non* sono (necessariamente) proporzionali a quelle degli archi del grafo (che sono, segmenti di retta).
Per rispondere alle due domande occorre elencare sistematicamente *tutti* i percorsi, che non passino più volte per uno stesso punto, tra n5 e n3:

PERCORSO da n5 a n3	LUNGHEZZA
[n5, n4, n1, n2, n3]	3+1+2+2=8
[n5, n4, n1, n3]	3+1+5=9
[n5, n4, n2, n3]	3+4+2=9
[n5, n4, n2, n1, n3]	3+4+2+5=14

L1, K1, L2, K2 seguono immediatamente.

ESEMPIO 2
L'ufficio tecnico di un piccolo comune deve scegliere dove piazzare dei nuovi lampioni.
Il paese di cui si parla può essere pensato come un insieme di piazzette collegate da strade, ovvero come un grafo con archi non diretti e non pesati, dove i nodi sono le piazze e gli archi sono le strade. Il grafo è descritto dal seguente elenco di termini:

arco(n1,n2) arco(n2,n3) arco(n3,n1)
arco(n4,n1) arco(n4,n2) arco(n4,n5)

Ogni lampione illumina:
la piazza in cui è collocato,
le strade da essa uscenti
le piazze direttamente collegate alla piazza in cui si trova il lampione.
(es. un lampione in n1 illumina la piazza n1, le strade n1-n2 n1-n3 n1-n4 e le piazze n2,n3,n4)

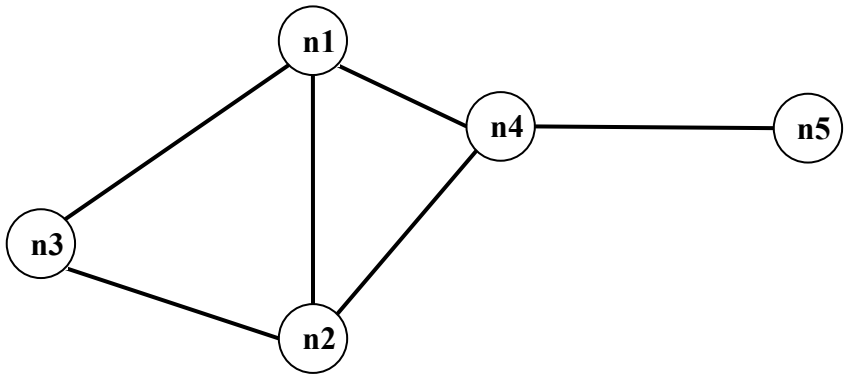
Trovare:
- il numero minimo di lampioni che consente di illuminare tutte le piazze del paese;
- la lista delle piazze (cioè dei nodi del grafo) su cui collocare tali lampioni, in modo che nessuna piazza sia illuminata da due lampioni.

N.B.
la lista deve avere gli elementi in ordine crescente (n1<n2< …<n5).

SOLUZIONE

numero minimo di lampioni	2
lista delle piazze	[n3,n5]

COMMENTI ALLA SOLUZIONE
Il grafo è il seguente.



Un metodo risolutivo generale è: generare tutti i sottoinsiemi di vertici, e per ciascuno verificare se permette di "illuminare" tutto il grafo; tra tutti quelli che soddisfano tale requisito, prendere quello/quelli con minor numero di elementi.
Questo metodo, può rapidamente diventare impraticabile: bastano già 5 nodi a renderlo difficoltoso. Tuttavia, poiché occorre individuare un sottoinsieme con minor numero di elementi che soddisfa il requisito, è ovvio che conviene generare ed esaminare i sottoinsiemi per cardinalità crescente. Il metodo quindi diventa:

1. verificare se esiste un vertice che da solo "illumina" tutto il grafo;
2. generare tutte le coppie e verificare se ne esiste una che "illumina" il grafo;
3. generare tutte le triple e verificare se ne esiste una che "illumina" il grafo;
   etc

Inoltre, in molti casi, semplici osservazioni sulla *topologia* del grafo possono portare immediatamente alla soluzione. Nel caso in esame, è chiaro che per coprire n5, la soluzione deve contenere o n4 oppure n5.
Se si pone un lampione in n4, tutte le piazze risultano illuminate tranne n3. Quindi con un lampione in n3 ed uno in n4 si illumina tutto il paese. Ciò consente di rispondere al primo quesito: il numero minimo di lampioni per illuminare il paese è 2.
Si osservi, tuttavia che ponendo i due lampioni in n3 ed n4, le piazze n1 ed n2 sono illuminate da due diversi lampioni. Esiste un'altra coppia di piazze in cui posizionare i lampioni in modo da illuminare tutte le piazze ed evitare che una piazza sia illuminata da due lampioni? Proviamo a posizionare un lampione in n5 invece che in n4. Tale lampione illumina n4 ed n5. Per illuminare le restanti 3 piazze, si può posizionare un secondo lampione in n3: esso illumina n1, n2 ed n3, ma non n4 ed n5. Quindi la soluzione al secondo quesito è la lista [n3,n5]

ESEMPIO 3
Paolo studia nella Repubblica di Indastria, una piccola nazione poco conosciuta ma molto moderna. Le principali città di Indastria sono collegate tra loro da voli in elicottero. Paolo abita nella città di Sapienza (dove ha sede l'università!) ma nel tempo libero vorrebbe visitare anche altre città, cercando di spendere il meno possibile.
Per descrivere i voli tra le città di Indastria, si usano dei termini così definiti:

volo(<partenza>,<destinazione>,<costo>)

Nel mese corrente, i voli disponibili sono i seguenti:

volo(Laborilla,Romantika,199) volo(Sapienza,Musa,250)
volo(Musa,Sapienza,300) volo(Musa,Romantika,280)
volo(Laborilla,Vecchioborgo,420) volo(Burovilla,Vecchioborgo,300)
volo(Romantika,Vecchioborgo,210) volo(Burovilla,Musa,310)

volo(Musa,Laborilla,450) volo(Sapienza,Laborilla,300)
volo(Musa,Burovilla,220)

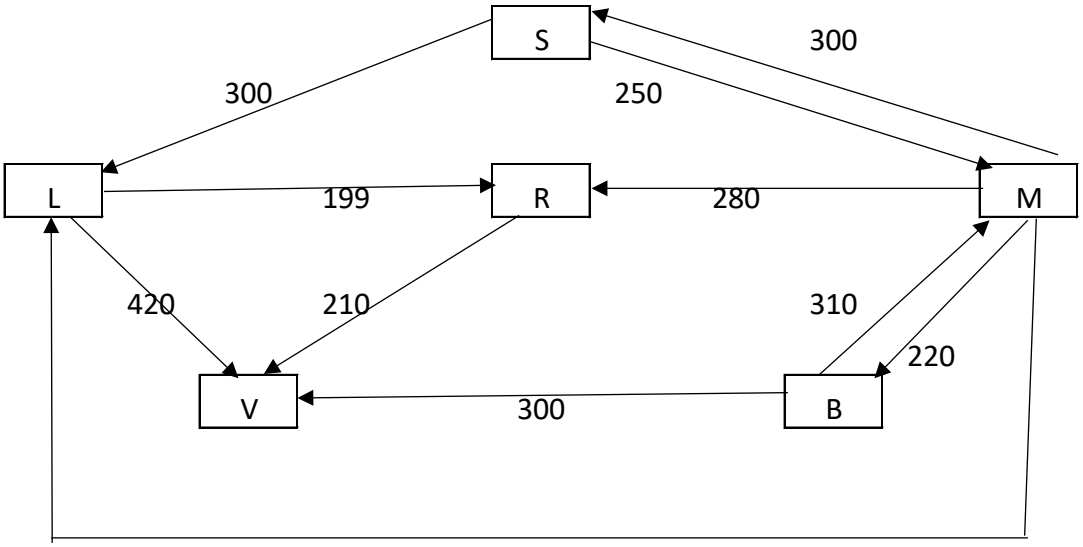
Un viaggio tra due città può essere descritto da una lista che contiene, in ordine, le città attraversate dal viaggio, compresa quella di partenza e quella di arrivo. Ad esempio un viaggio da Musa a Romantika che attraversa Sapienza e Laborilla è descritto dalla lista [Musa,Sapienza,Laborilla,Romantika].
Paolo approfitta delle attese tra un volo e il successivo per visitare la città in cui fa scalo, ma non vuole visitare più di una volta una stessa città. Aiuta Paolo a pianificare i suoi viaggi, trovando:

1. Il nome C della città che ha il maggior numero di collegamenti, in arrivo o in partenza
2. La lista L1 del viaggio per andare da Burovilla a Romantika facendo un viaggio che attraversa almeno altre due città (diverse da Burovilla a Romantika), ma non Sapienza
3. La lista L2 del viaggio per andare da Sapienza a Vecchioborgo spendendo il meno possibile

SOLUZIONE

C	Musa
L1	[Burovilla,Musa,Laborilla,Romantika]
L2	[Sapienza,Laborilla,Romantika,Vecchioborgo]

COMMENTI ALLA SOLUZIONE
La situazione descritta dal testo del problema si può rappresentare mediante un grafo, orientato e pesato, in cui le città sono i nodi e i voli sono gli archi. Il grafo è orientato perché i voli hanno una sola direzione e il peso di ciascun arco rappresenta il costo del volo. Il disegno del grafo corrispondente all'elenco dei termini nel testo del problema è il seguente (nei nodi sono indicate solo le iniziali dei nomi delle città):



Usando l'interpretazione mediante grafo, il primo quesito consiste nell'individuare il nodo per il quale la somma tra il grado di entrata e il grado di uscita ha valore massimo. Ispezionando la figura si vede facilmente che tale nodo è Musa che ha 2 archi entranti e 4 uscenti.
Il secondo e il terzo quesito ci chiedono individuare un viaggio, che non deve passare due volte nella stessa città. Ciascun viaggio, quindi, corrisponde ad un cammino semplice sul grafo.
Per risolvere il secondo quesito, si devono esaminare i cammini semplici da Burovilla a Romantika. Da Burovilla partono due voli, per Vecchioborgo e per Musa. Poiché Vecchioborgo non ha archi uscenti, il cammino che cerchiamo deve proseguire per forza in Musa. Da Musa si può andare direttamente a Romantika, ma tale viaggio non può essere quello chiesto, perché non attraversa almeno due città diverse da quella di partenza e quella di arrivo. In alternativa, da Musa si può andare a Romantika attraversando Laborilla oppure attraversando prima Sapienza e poi Laborilla.

Poiché ci viene richiesto di individuare un viaggio che non attraversa Sapienza, si ha L1=[Burovilla,Musa,Laborilla,Romantika].
Per risolvere il terzo quesito, si devono esaminare tutti i cammini semplici da Sapienza a Vecchioborgo e scegliere quello per il quale la somma dei pesi degli archi è minore.
Si parte da Sapienza e si elencano tutte le possibili prime destinazioni (cammini di lunghezza 1):

- [Sapienza,Laborilla]
- [Sapienza,Musa]

Si allungano i cammini considerando tutti gli archi che escono dai nodi terminali di ciascuno dei cammini:

- Da Laborilla escono archi verso Vecchioborgo e Romantika.
- Da Musa escono archi verso Laborilla, Romantika e Burovilla.

Per cui si generano i seguenti cammini di lunghezza 2:

- **[Sapienza,Laborilla,Vecchioborgo]**
- [Sapienza,Laborilla,Romantika]
- [Sapienza,Musa,Laborilla]
- [Sapienza,Musa,Romantika]
- [Sapienza,Musa,Burovilla]

È stato trovato un primo cammino che raggiunge la destinazione, evidenziato in **grassetto**.
Per tutti gli altri cammini, si ripete il processo di allungamento.

- Da Romantika esce un arco verso Vecchioborgo.
- Da Laborilla, escono archi verso Vecchioborgo e Romantika.
- Da Burovilla esce un arco verso Vecchioborgo.

Per cui si generano i seguenti cammini di lunghezza 3:

- **[Sapienza,Laborilla,Romantika,Vecchioborgo]**
- **[Sapienza,Musa,Laborilla,Vecchioborgo]**
- [Sapienza,Musa,Laborilla,,Romantika]
- **[Sapienza,Musa,Romantika,Vecchioborgo]**
- **[Sapienza,Musa,Burovilla,Vecchioborgo]**

Il processo di allungamento questa volta ha individuato ben 4 cammini di lunghezza 3 che raggiungono la destinazione. Resta un solo cammino candidato, che continuiamo ad allungare.
Da Romantika esce un arco verso Vecchioborgo. Quindi individuiamo un solo altro cammino di lunghezza 4 che giunge a destinazione:

- **[Sapienza,Musa,Laborilla,Romantika,Vecchioborgo]**

Non vi sono più cammini candidati, quindi il procedimento termina. A questo punto non ci resta che esaminare i cammini evidenziati in grassetto, e per ciascuno di essi calcolare il costo.
Riportiamo cammini e costi nella seguente tabella.

Lunghezza cammino	CAMMINO da Sapienza a Vecchioborgo	Costo
2	[Sapienza,Laborilla,Vecchioborgo]	720
3	[Sapienza,Laborilla,Romantika,Vecchioborgo]	709
3	[Sapienza,Musa,Laborilla,Vecchioborgo]	1120

3	[Sapienza,Musa,Romantika, Vecchioborgo]	740
3	[Sapienza,Musa,Burovilla,Vecchioborgo]	770
4	[Sapienza,Musa,Laborilla,Romantika,Vecchioborgo]	1109

Confrontando i costi riportati in tabella si determina la soluzione.

d) KNAPSACK

Illustriamo questa tipologia di problemi attraverso due esempi.

ESEMPIO 1

In un deposito di minerali esistono esemplari di vario peso e valore individuati da sigle di riconoscimento. Ciascun minerale è descritto mediante il seguente termine di nome tab avente tre argomenti:

tab(<sigla del minerale>,<valore in euro>,<peso in kg>).

Il deposito contiene quattro minerali:

tab(m1,15,35) tab(m2,19,46) tab(m3,14,25) tab(m4,10,12)

Disponendo di un piccolo motocarro con portata massima di 59 kg trovare la lista L delle sigle di due minerali diversi che siano trasportabili contemporaneamente con questo mezzo e che abbiano il massimo valore complessivo; calcolare inoltre questo valore V.

N.B. Nella lista, elencare le sigle in ordine (lessicale) crescente; per le sigle usate si ha il seguente ordine: m1<m2<m3< … .

SOLUZIONE

L	[m2,m4]
V	29

COMMENTI ALLA SOLUZIONE

Per risolvere il problema occorre considerare *tutte* le possibili *combinazioni* di due minerali diversi, il loro valore e il loro peso.

N.B. Le *combinazioni* corrispondono ai sottoinsiemi: cioè sono indipendenti dall'ordine; per esempio la combinazione "m1,m4" è uguale alla combinazione "m4,m1". Quindi per elencarle tutte (una sola volta) conviene costruirle sotto forma di liste i cui elementi sono ordinati, come richiesto dal problema.

Costruite le combinazioni occorre individuare quelle trasportabili (cioè con peso complessivo minore o eguale a 59) e tra queste scegliere quella di maggior valore.

COMBINAZIONI	VALORE	PESO	TRASPORTABILI
[m1,m2]	15+19=34	35+46=81	no
[m1,m3]	15+14=29	35+25=60	no
[m1,m4]	15+10=25	35+12=47	si
[m2,m3]	19+14=33	46+25=71	no
[m2,m4]	19+10=29	46+12=58	si
[m3,m4]	14+10=24	25+12=37	si

Dal precedente prospetto si deduce facilmente la soluzione.

N.B. Conviene elencare (costruire) prima tutte le combinazioni che iniziano col "primo" minerale, poi tutte quelle che iniziano col "secondo" minerale, e così via, in modo da essere sicuri di averle considerate tutte.

ESEMPIO 2

In un deposito di minerali esistono esemplari di vario peso e valore individuati da sigle di riconoscimento. Ciascun minerale è descritto mediante il seguente termine di nome tab avente tre argomenti:

tab(<sigla del minerale>,<valore in euro>,<peso in kg>).

Il deposito contiene sei minerali:

minerale(m1,39,58)	minerale(m2,42,64)	minerale(m3,40,65)
minerale(m4,38,59)	minerale(m5,37,61)	minerale(m6,42,62)

I minerali possono essere spostati con carrelli di diversa portata su cui si possono mettere tre esemplari (diversi).

- Disponendo di un carrello con portata massima di 180 kg, trovare la lista L1 delle sigle di tre minerali diversi che siano trasportabili contemporaneamente e che abbiano il massimo valore complessivo.
- Disponendo di un carrello con portata massima di 185 kg, trovare la lista L2 delle sigle di tre minerali diversi che siano trasportabili contemporaneamente e che abbiano il massimo valore complessivo.
- Disponendo di un carrello con portata massima di 200 kg, trovare la lista L1 delle sigle di tre minerali diversi che siano trasportabili contemporaneamente e che abbiano il massimo valore complessivo.

N.B. Nella lista, elencare le sigle in ordine (lessicale) crescente; per le sigle usate si ha il seguente ordine: m1<m2<m3< …

L1	
L2	
L3	

SOLUZIONE

L1	[m1,m4,m6]
L2	[m1,m2,m6]
L3	[m2,m3,m6]

COMMENTI ALLA SOLUZIONE

In generale, in problemi di questo tipo, occorre considerare *tutte* le possibili *combinazioni* di tre minerali diversi; in questo caso occorre inoltre, per ognuna, determinare il valore, il peso e il carrello più "piccolo" che la può trasportare.

N.B. Le *combinazioni* corrispondono ai sottoinsiemi: cioè sono indipendenti dall'ordine; per esempio la combinazione "m1, m2, m3" è uguale alla combinazione "m3, m2, m1". Quindi per elencarle tutte (una sola volta) conviene costruirle sotto forma di liste i cui elementi sono ordinati come richiesto dal problema.

COMBINAZIONE	VALORE	PESO	CARRELLO MIN.
[m1,m2,m3]	121	187	200
[m1,m2,m4]	119	181	185

[m1,m2,m5]	118	183	185	
[m1,m2,m6]	123	184	185	L2
[m1,m3,m4]	117	182	185	
[m1,m3,m5]	116	184	185	
[m1,m3,m6]	121	185	185	
[m1,m4,m5]	114	178	180	
[m1,m4,m6]	119	179	180	L1
[m1,m5,m6]	118	181	185	
[m2,m3,m4]	120	188	200	
[m2,m3,m5]	119	190	200	
[m2,m3,m6]	124	191	200	L3
[m2,m4,m5]	117	184	185	
[m2,m4,m6]	122	185	185	
[m2,m5,m6]	121	187	200	
[m3,m4,m5]	115	185	185	
[m3,m4,m6]	120	186	200	
[m3,m5,m6]	119	188	200	
[m4,m5,m6]	117	182	185	

Costruite le combinazioni, occorre individuare, per ogni carrello, quella di maggior valore.

e) PIANIFICAZIONE

In generale si parla di un progetto che per essere realizzato è suddiviso in varie attività A1, A2 ecc..
Per ogni attività, si conosce in quanti giorni viene svolta e quante sono le persone coinvolte.
Le attività devono *succedersi opportunamente* nel tempo perché, per esempio, una attività utilizza il prodotto di altre: quindi le *priorità* sono descritte con coppie di sigle. Ad esempio, la priorità [A1,A2] indica che l'attività A2 potrà essere iniziata solo dopo il completamento dell'attività A1.
Due sono le tecniche da usare: il diagramma di Pert e il diagramma di Gantt.
Illustriamo questa tipologia di problemi attraverso tre esempi.

ESEMPIO 1

La tabella che segue descrive le attività di un progetto (indicate rispettivamente con le sigle A1, A2, ...), riportando per ciascuna di esse il numero di giorni necessari per completarla.

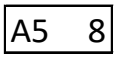
Attività	Giorni
A1	2
A2	3
A3	4
A4	1

Le priorità tra le attività del progetto sono: [A1,A2], [A1,A3], [A2,A4], [A3,A4]
la prima attività è la A1 (non è mai presente in seconda posizione) e l'ultima attività è la A4 (non è mai presente in prima posizione). Per ogni altra attività si individuano le precedenze:

A1 non ha precedenze
[A1,A2] indica che A2 deve essere eseguita appena terminata A1
[A1,A3] indica che A3 deve essere eseguita appena terminata A1;
A3 inizia nello stesse giorno in cui inizia A2.
[A2,A4], [A3,A4] indicano che A4 inizierà appena terminata la più lunga (in giorni) tra le attività A2 e A3.
A4 è l'attività che completa il progetto.

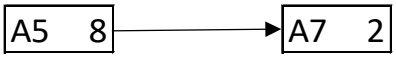
La sequenza delle priorità è rappresentata graficamente utilizzando la simbologia seguente :

- un'attività e i giorni necessari per completarla sono rappresentati da un rettangolo con all'interno : a sinistra la sigla dell'attività e a destra il numero dei giorni .

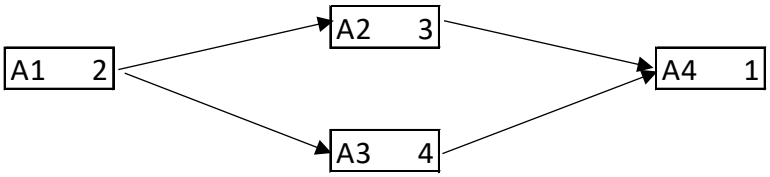


- la priorità viene rappresentata da una freccia che parte dall'attività iniziale e arriva all'attività finale

Es L'attività A5 si completa in 8 giorni e l'attività A7 in 2 giorni. La priorità è [A5,A7]



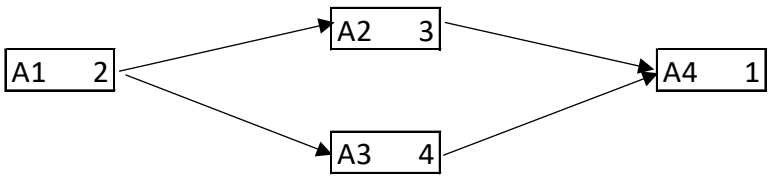
Il grafico che si ottiene è detto ***diagramma di Pert.***



Per trovare il numero minimo N di giorni necessari per completare il progetto rispettando le priorità, servirà calcolare la somma dei giorni necessari scegliendo l'attività più lunga, quando si valutando quelle che possono essere svolte in contemporanea. Nel nostro caso, il numero minimo di giorni necessari per completare il progetto sarà: 2 (giorni per completare A1) + 4 (giorni per completare la più lunga attività fra A2 e A3) + 1 (giorni per completare A4) = 7.
Tutto questo si rappresenta aggiungendo sotto il diagramma di Pert la sequenza

numero giorni attività A1 ……. numero giorni attività più lunga ……. numero giorni attività A4

Nel nostro esempio abbiamo



2 …… 4 ……1

ESEMPIO 2

Alcuni ragazzi decidono di costruire un ipertesto multimediale sugli avvenimenti significativi della loro regione per la prossima stagione turistica. Per organizzare il progetto, dividono il lavoro in singole attività e, per ciascuna di queste stabiliscono quanti di loro devono partecipare e stimano il tempo per portarla a conclusione. La tabella che segue descrive le attività (indicate rispettivamente con le sigle A1, A2, A3, …), riportando per ciascuna di esse il numero di ragazzi assegnato e il numero di giorni necessari per completarla.

ATTIVITÀ	RAGAZZI	GIORNI
A1	6	2
A2	4	2
A3	3	3
A4	6	2
A5	4	2
A6	5	1

N.B. Ai fini del problema non è importante conoscere la descrizione delle singole attività.

Le attività devono *succedersi opportunamente* nel tempo perché, per esempio, una attività utilizza il prodotto di altre: quindi esistono delle *priorità* descritte con coppie di sigle; ogni coppia esprime il fatto che l'attività associata alla sigla di destra (detta *successiva*) può iniziare solo quando l'attività associata alla sigla di sinistra (detta *precedente*) è terminata. Ovviamente se una attività ha più precedenti, può iniziare solo quando tutte le precedenti sono terminate.
In questo caso le priorità sono:

[A1,A2], [A1,A3], [A2,A4], [A3,A4], [A4,A5], [A5,A6].

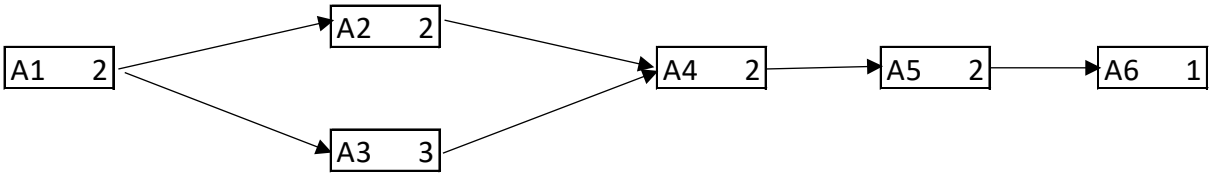
Trovare il numero N di giorni necessari per completare il progetto, tenuto presente che alcune attività possono essere svolte in parallelo e che ogni attività *deve* iniziare prima possibile (nel rispetto delle priorità). Inoltre, trovare il numero massimo RM di ragazzi che lavora contemporaneamente al progetto.

SOLUZIONE

N	10
RM	7

COMMENTI ALLA SOLUZIONE

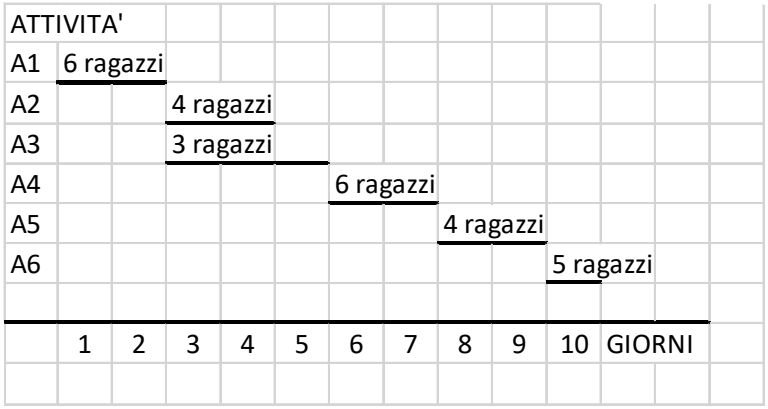
Per prima cosa, dai dati sulle priorità occorre tracciare il diagramma di Pert che ha come nodi le attività e come frecce le precedenze: indica visivamente la dipendenza "logica" tra le attività, quindi come si devono susseguire nel tempo.



2 ……..…… 3 ……. 2 …… 2 ……..1

Il progetto si svolge 10 giorni: 2 (giorni per completare A1) + 3 (giorni per completare la più lunga attività fra A2 e A3) + 2 (giorni per completare A4) + 2 (giorni per completare A5) + 1 (giorno per completare A6) .

Per rispondere alla seconda domanda si realizza un secondo diagramma detto ***diagramma di Gantt.***
Questo riporta sull'asse verticale le attività (dall'alto verso il basso), sugli assi orizzontali il tempo, in questo caso misurato in giorni. Su ogni asse orizzontale (parallelo a quello dei tempi e in corrispondenza a una attività) è sistemato un segmento che indica l'inizio e la durata della corrispondente attività (e il numero di ragazzi che devono svolgerla).
Così, per esempio, l'attività A1 inizia il giorno 1 e dura due giorni; quando è terminata, il giorno 3 possono iniziare le attività A2 e A3 (che quindi si svolgono parzialmente in parallelo). L'attività A4 può iniziare solamente quando è terminata sia A3 sia A2.



Il Gantt conferma che il progetto dura 10 giorni e mostra che il numero massimo di ragazzi al lavoro contemporaneamente è 7 (nei giorni 3 e 4).

ESEMPIO 3
La tabella che segue descrive le attività di un progetto (indicate rispettivamente con le sigle A1, A2, …), riportando per ciascuna di esse il numero di persone assegnato e il numero di giorni necessari per completarla.

ATTIVITÀ	PERSONE	GIORNI
A1	6	2
A2	3	3
A3	2	4
A4	6	1
A5	2	3
A6	2	4
A7	3	2
A8	2	4
A9	5	1

Le priorità tra le attività sono:

[A1,A2], [A1,A3], [A1,A4], [A2,A5], [A3,A8], [A7,A9],
[A4,A6], [A6,A9], [A5,A7], [A8,A9].

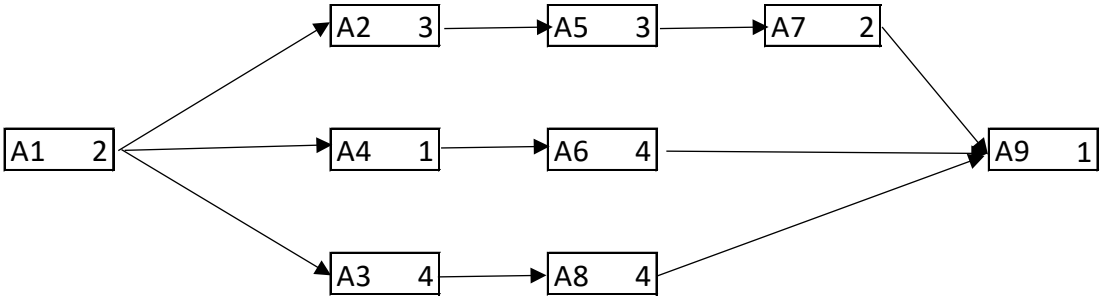
Trovare il numero N di giorni necessari per completare il progetto, tenuto presente che alcune attività possono essere svolte in parallelo e che ogni attività *deve* iniziare prima possibile (nel rispetto delle priorità). Inoltre, determinare PM: il *numero massimo* di persone che lavorano contemporaneamente al progetto.
(N.B. PM è anche il *numero minimo* di persone contemporaneamente disponibili necessarie per attuare il progetto così pianificato).

SOLUZIONE

N	11
PM	11

COMMENTI ALLA SOLUZIONE

Per prima cosa, dai dati sulle priorità occorre disegnare il diagramma di Pert, cioè il grafo che ha come nodi le attività e come frecce le precedenze.



2…… 8 …….. 1

Il progetto si svolge in11 giorni: 2 (giorni per completare A1) + 8 (giorni per completare le attività A2,A5,A7 e A3,A8) + 1 (giorno per completare A9) .

Per la seconda risposta compiliamo il diagramma di Gantt tenendo conto che l'attività A1 inizia il giorno 1 e dura due giorni; quando è terminata, il giorno 3 possono iniziare le attività A2, A3 e A4 (che quindi si svolgono parzialmente in parallelo); inoltre l'attività A7, come altro esempio, può iniziare solamente quando è terminata la A5.

ATTIVITA'													
A1	6 persone												
A2			3 persone										
A3			2 persone										
A4			6 persone										
A5						2persone							
A6				2 persone									
A7									3 persone				
A8								2 persone					
A9											5 persone		
	1	2	3	4	5	6	7	8	9	10	11		GIORNI

Il Gantt conferma che il progetto dura 11 giorni e mostra che il numero *massimo* di persone al lavoro contemporaneamente è 11 (il terzo giorno): quindi per realizzare il progetto occorre almeno la disponibilità contemporanea di 11 persone.

f) CRITTOGRAFIA

In generale crittare (o criptare) un messaggio significa trasformarlo in una serie di simboli di difficile (se non impossibile) comprensione utilizzando certe regole. Solo chi le conosce è in grado di ricostruire il messaggio originale.

L'insieme delle regole utilizzate prende il nome di algoritmo di crittazione.

CIFRARIO A SOSTITUZIONE MONOALFABETICA GENERICO

È costituito da una tabella di conversione attraverso la quale si trasforma ogni simbolo del messaggio in chiaro in un altro. La particolare tabella usata è detta *chiave di crittazione*. Ad esempio, con la seguente tabella di conversione (o chiave di crittazione):

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
W	X	Y	U	V	N	K	L	M	O	P	Q	R	S	T	Z	D	E	F	A	B	C	G	H	I	J

(ovvero la A diventa una W, la B una X, etc.)

Esempio. Utilizzando la tabella:

1. crittare la parola NASO
2. decrittare il messaggio "ZWQMT UM FMVSW"
3. crittare la parola MELA utilizzando la seguente regola:
   per ogni lettera applicare quattro volte la tabella
   es. la lettera C si trasforma nella lettera R perché C – Y—I—M—R

SOLUZIONE

1	SWFT
2	PALIO DI SIENA
3	CIBP

COMMENTI ALLA SOLUZIONE

1)Dalla tabella segue N—S, A—W, S—F, O—T è crittata in SWFT.

2)Sempre utilizzando la tabella leggiamo la lettera nella seconda riga (verso il basso); il messaggio in chiaro sarà formato dalle corrispondenti lettere nella prima riga.

P	A	L	I	O		D	I		S	I	E	N	A
Z	W	Q	M	T		U	M		F	M	V	S	W

3)Presa una lettera applichiamo quattro volte la tabella:

	1	2	3	4
M	R	E	V	**C**
E	V	C	Y	**I**
L	Q	D	U	**B**
A	W	G	K	**P**

IL CIFRARIO DI CESARE

È ancora un cifrario a sostituzione monoalfabetica in cui ogni lettera del testo in chiaro è sostituita nel testo cifrato dalla lettera che si trova un certo numero di posizioni dopo nell'alfabeto. Questi tipi di cifrari sono detti anche **cifrari a sostituzione** o **cifrari a scorrimento** a causa del loro modo di operare: la sostituzione avviene lettera per lettera, scorrendo il testo dall'inizio. Il numero di posti di scorrimento è detto chiave. Ad esempio per la chiave 5 (la a "scorre" in avanti di 5 posti e diventa f) abbiamo

	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**	**n**	**o**	**p**	**q**	**r**	**s**	**t**	**u**	**v**	**w**	**x**	**y**	**z**
5	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e

Il testo «olimpiadi di problem solving» viene cifrato come «tqnrunfin in uwtgqjr xtqansl»
In modo analogo il testo cifrato con tale chiave «knsj xjyynrfsf» viene decrittato come «fine settimana»
Al variare della chiave una stessa parola viene crittata in modo diverso.
Ad esempio "roma" è "tqoc" in chiave 2 e "ligu" in chiave 20.

N.B. In un alfabeto di 26 lettere esistono 25 cifrari di Cesare diversi.

chiave 1 a scorre in b
chiave 2 a scorre in c
chiave 3 a scorre in d
……
chiave 25 a scorre in z
chiave 26 a scorre in a per cui il testo in chiaro verrebbe crittato in sé stesso.

Esempio.
Usando un cifrario di Giulio Cesare:

1. crittare il messaggio "STAZIONE DI MILANO" con chiave 3;
2. decrittare la parola "GTQTLSF "sapendo che è stata utilizzata la chiave 5;
3. trovare la chiave con la quale "MELA" è stata crittata in "UMTI"

SOLUZIONE

1	VWDCLRQH GL PLODQR
2	BOLOGNA
3	8

COMMENTI ALLA SOLUZIONE

1)Costruita la chiave 3

	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**	**n**	**o**	**p**	**q**	**r**	**s**	**t**	**u**	**v**	**w**	**x**	**y**	**z**
3	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c

segue

S	T	A	Z	I	O	N	E		D	I		M	I	L	A	N	O
V	W	D	C	L	R	Q	H		G	L		P	L	O	D	Q	R

2) Utilizziamo la chiave 5

	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**	**n**	**o**	**p**	**q**	**r**	**s**	**t**	**u**	**v**	**w**	**x**	**y**	**z**
5	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e

cercando ogni lettera nella seconda riga e scrivendo la corrispondente nella prima riga

G	T	Q	T	L	S	F
B	**O**	**L**	**O**	**G**	**N**	**A**

3)Per rispondere alla domanda bisogna cercare il cifrario in cui la A scorre in I.

A	→						I
B	C	D	E	F	G	H	
1	2	3	4	5	6	7	8

La risposta è 8 e con la corrispondente chiave

	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**	**n**	**o**	**p**	**q**	**r**	**s**	**t**	**u**	**v**	**w**	**x**	**y**	**z**
8	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h

possiamo verificare che MELA si critta in UMTI.

N.B. Per individuare la chiave si può partire da una qualsiasi lettera e contare di quanti posti scorre per giungere alla sua corrispondente.

CIFRARIO POLIALFABETICO DI VIGENÈRE

Mettendo ordinatamente insieme tutte e 26 le chiavi otteniamo la tabella di Vigenère riprodotta qui sotto:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
B	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
C	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
D	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
E	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d
F	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
G	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
H	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g
I	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h
J	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
K	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j
L	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k
M	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l
N	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m
O	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n
P	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
Q	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
R	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
S	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
T	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
U	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
V	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
W	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
X	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
Y	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
Z	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y

Questa permette la realizzazione de seguente algoritmo di crittazione a sostituzione *polialfabetica*:

Si sceglie una parola chiamata chiave
Si prende la prima lettera del messaggio e la prima della chiave

Queste due lettere vengono usate come coordinate cartesiane nel senso che individuano una riga e una colonna della tabella; all'incrocio si trova la lettera da sostituire per crittare il messaggio.
Per le altre lettere del messaggio si procede allo stesso modo ma utilizzando le successive lettere della chiave; siccome in generale la chiave è più corta del messaggio, la stessa chiave verrà usata ripetutamente fino a completare la crittazione del messaggio.
Con questo metodo lettere uguali possono avere cifrature differenti

ESEMPIO 1 Usando come chiave la parola TRE, crittare:

1) STELLA

2) LA SOLITUDINE DEI NUMERI PRIMI

Scrivere le risposte ricordando che nel caso di più parole, ogni parola deve distanziarsi dall'altra di un SOLO spazio.

SOLUZIONE

1	LKIECE
2	ER WHCMMLHBEI WVM GLQXIM IIMFZ

COMMENTI ALLA SOLUZIONE

1. Utilizzando la tabella di Vigenère e seguendo le indicazioni per la crittazione abbiamo

		S				T				E				L				L				A
T		**L**			R		**K**			E		**I**			T		E			R		C				E		E

Dunque;

testo in chiaro	S	T	E	L	L	A
chiave	T	R	E	T	R	E
testo crittato	**L**	**K**	**I**	**E**	**C**	**E**

2. In questo caso per crittare dobbiamo prima eliminare gli spazi tra le parole trasformando la frase nella lista

LASOLITUDINEDEINUMERIPRIMI

e ripetere più volte la chiave TRE.

L	A	S	O	L	I	T	U	D	I	N	E	D	E	I	N	U	M	E	R	I	P	R	I	M	I
T	R	E	T	R	E	T	R	E	T	R	E	T	R	E	T	R	E	T	R	E	T	R	E	T	R

Utilizzando come sopra la tabella di Vigenère otteniamo il seguente messaggio crittato

ER WHCMMLHBEI WVM GLQXIM IIMFZ

Per decrittare un messaggio, si considera la prima lettera del messaggio crittato e la prima lettera della chiave; la lettera della chiave si usa per individuare la riga nella tavola di Vigenère. Individuata la riga si cerca in essa l'elemento corrispondente alla lettera del messaggio crittato. Individuata la colonna, si considera la lettera corrispondente sarà la prima lettera del messaggio decrittato. Si procede in modo analogo per il resto delle lettere.

ESEMPIO 2. Usando come chiave la parola TRE, decrittare:

1) IIMFZ

2) HII WZIVZ MG GMTQDT

Scrivere le risposte rispettivamente nella prima e seconda riga della tabella sottostante.
Nel caso di più parole, ogni parola deve distanziarsi dall'altra di un SOLO spazio.

SOLUZIONE

1	PRIMI
2	ORE DIECI IN PIAZZA

COMMENTI ALLA SOLUZIONE

1. Utilizzando la tabella di Vigenère e seguendo le indicazioni per la decrittazione abbiamo

		P				R				I				M				I
T		I			R		I			E		M			T		F			R		Z

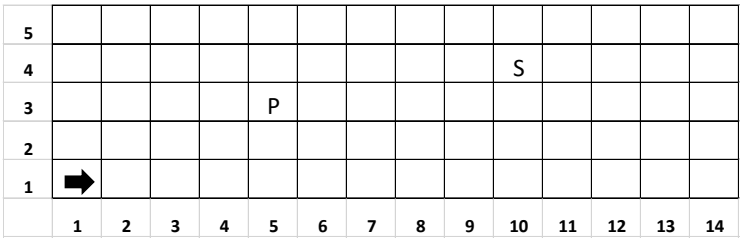
2. Si inizia costruendo la lista HIIWZIVZMGGHTQD ottenuta eliminando gli spazi tra le parole. Successivamente per mezzo della tabella di Vigenère si decritta ottenendo il messaggio in chiaro separando le parole.

testo crittato	H	I	I	W	Z	I	V	Z	M	G	G	M	T	Q	D	T
chiave	T	R	E	T	R	E	T	R	E	T	R	E	T	R	E	T
messaggio in chiaro	O	R	E	D	I	E	C	I	I	N	P	I	A	Z	Z	A

g) ROBOT E AUTOMI

ROBOT CLASSICO

In un foglio a quadretti è disegnato un "campo di gara", con 14 quadretti in orizzontale e 5 in verticale (vedi figura).



Ogni casella è individuata da due numeri (interi); per esempio la casella contenente P è individuata da essere nella quinta colonna (da sinistra) e nella terza riga (dal basso): brevemente si dice che ha *coordinate* [5,3]; la prima coordinata (in questo caso 5) si dice *ascissa* e la seconda (in questo caso 3) si dice *ordinata*.
Le coordinate della casella contenente S sono [10,4] e di quella contenente la freccia sono [1,1].
La freccia può essere pensata come un robot, in questo caso rivolto verso destra; lo stato del robot può quindi essere individuato da tre "valori": due per le coordinate della casella che occupa e uno per indicare il suo orientamento. Per quest'ultimo si possono usare i simboli della rosa dei venti: E, S, W, N: per indicare che il robot è rivolto, rispettivamente, a *destra*, in *basso*, a *sinistra*, in *alto* (con riferimento a chi guarda il foglio); lo stato del robot, rappresentato dalla freccia nella figura è [1,1,E].
Il robot può eseguire tre tipi di comandi:

- ruotare di 90 gradi in senso *orario*: comando **o**;
- ruotare di 90 gradi in senso *antiorario*: comando **a**;
- avanzare di una casella (nel senso della freccia, mantenendo l'orientamento): comando **f**.

Per indicare una rotazione di 180° useremo SEMPRE il *doppio comando* **a,a**.
Questi comandi possono essere concatenati in sequenze in modo da permettere al robot di compiere vari percorsi.
La sequenza di comandi descritta dalla lista [f,f,f,f,a,f,f] fa spostare il robot dalla posizione e orientamento iniziali mostrati in figura fino alla casella P; le caselle via via occupate (quella di partenza e quella di arrivo comprese) sono quelle della lista:

[[1,1],[2,1],[3,1],[4,1],[5,1],[5,2],[5,3]].

Stessa casella di arrivo si raggiunge con la lista di comandi [a,f,f,o,f,f,f,f], ma il percorso è diverso ed è descritto dalla lista

[[1,1],[1,2],[1,3],[2,3],[3,3],[4,3],[5,3]].

Inoltre, nel primo caso lo stato l'orientamento finale del robot è verso l'alto (stato [5,3,N]), mentre nel secondo caso l'orientamento finale è verso destra (stato [5,3,E]).

ESEMPIO 1
In un campo di gara il robot si trova nella casella [2,8]) con direzione West e deve eseguire la seguente lista di comandi [o,f,f,o,f,f,o,f,f].
Determinare:

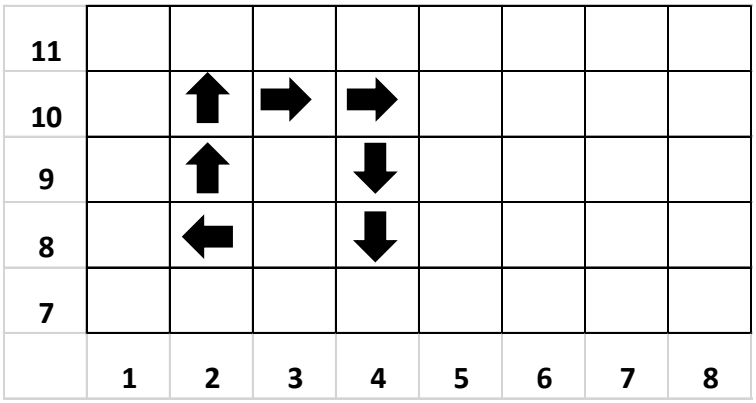
1. lo stato S1 in cui si trova il robot <u>prima</u> di aver eseguito tutti i comandi
2. lo stato S2 in cui si trova il robot <u>dopo</u> aver eseguito tutti i comandi

SOLUZIONE

S1	[2,8,W]
S2	[4,8,S]

COMMENTI ALLA SOLUZIONE
La direzione è indicata con le iniziali delle parole Nord (alto), Sud (basso), Est (destra) e West (sinistra).
La lista di comandi è [o,f,f,o,f,f,o,f,f]. La posizione iniziale è (2,8) e la direzione iniziale del robot è West. Quindi S1 è la lista [2,8,W].

Per determinare S2, è conveniente visualizzare il percorso, come nella figura che segue (che mostra solo parzialmente il campo di gara, con il valore delle coordinate).



Osservando la figura è semplice determinare la sequenza di comandi che fa compiere tale percorso. Si deve prestare attenzione all'orientamento del robot. Inizialmente il robot si trova in [2,8] con direzione West, ovvero ha stato [2,8,W]. Il primo comando modifica la direzione, portandola a Nord, ovvero trasforma lo stato in [2,8,N]. Il secondo comando fa percorrere un passo lungo la direzione del robot, e quindi lo stato diviene [2,9,N].
Ragionando in modo analogo, si ricostruiscono tutti i movimenti, riassunti nella seguente tabella che mostra, per ogni comando, l'evoluzione dello stato del robot.

Stato di partenza	Comando	Stato di arrivo
[2,8,W]	o	[2,8,N]
[2,8,N]	f	[2,9,N]
[2,9,N]	f	[2,10,N]
[2,10,N]	o	[2,10,E]
[2,10,E]	f	[3,10,E]
[3,10,E]	f	[4,10,E]
[4,10,E]	o	[4,10,S]
[4,10,S]	f	[4,9,S]
[4,9,S]	f	[4,8,S]

ROBOT con comando r

Il un nuovo modello di robot, oltre a eseguire movimenti tramite i comandi **o**, **a** ed **f**, che ben conosciamo, ha una nuova caratteristica: la capacità di ripetere un determinato numero di volte delle sotto-liste di comandi! In dettaglio, il robot oltre a **o**, **a** ed **f**, esegue il nuovo comando **r**: questo comando è seguito dal *numero di ripetizioni*, poi da una sequenza di comandi chiamata *corpo* e infine dal simbolo **|** che indica la fine del *corpo*

- un esempio è **r5f,a,f|** in cui 5 è il *numero di ripetizioni* e **f,a,f** è il *corpo*
- l'esecuzione di **r** è equivalente a eseguire i comandi che formano il *corpo* di **r**, un numero di volte pari al *numero di ripetizioni*

Ad esempio, se al robot viene data la lista di comandi [**f,r3f,f,a|,a,f**], il robot si comporta come segue:

1. esegue **f**

2. esegue *tre volte* **r**, ovvero
    a. esegue **f**
    b. esegue **f**
    c. esegue **a**
    d. esegue **f**
    e. esegue **f**
    f. esegue **a**
    g. esegue **f**
    h. esegue **f**
    i. esegue **a**
3. esegue **a**
4. esegue **f**

ESEMPIO 2
Flavia posiziona il suo nuovo robot nella casella [8,13] di un campo, con direzione E. Attiva il robot con la seguente lista di comandi L=[**ar2aff|fr4fofa|f**].
Quali saranno la posizione P1 e la direzione D1 dopo aver eseguito il primo dei due comandi **r**? Scrivi P1 sotto forma di lista di coordinate e D1 come singola lettera nella tabella sottostante.
Quali saranno la posizione P2 e la direzione D2 dopo aver eseguito l'intera lista di comandi? Scrivi P2 sotto forma di lista di coordinate e D2 come singola lettera nella tabella sottostante.

SOLUZIONE

P1	[6,11]
D1	S
P2	[2,5]
D2	S

COMMENTI ALLA SOLUZIONE
Per risolvere il problema possiamo trasformare la lista L in una lista priva del comando r, e quindi compatibile con il vecchio modello di robot. Fatto ciò, il problema si risolve come i tanti problemi con il robot "classico" affrontati nelle gare di allenamento.
Per trasformare L facciamo in questo modo: esaminiamo L da sinistra verso destra. Ogni volta che troviamo un comando r, lo eliminiamo (insieme a tutti i caratteri che lo seguono fino a | compreso) e al suo posto inseriamo tante copie del corpo del comando quanto è il valore del numero di ripetizioni del comando. Usando questo procedimento, la lista L si trasforma nella lista

M=[**aaffaffffofafofafofafof**].

Eseguendo M con un robot classico si trovano le risposte ai quesiti.

inizio		[	8	,	13	,	E	]			
	a	[	8	,	13	,	N	]			
1	a	[	8	,	13	,	W	]			
1	f	[	7	,	13	,	W	]			
1	f	[	6	,	13	,	W	]			
2	a	[	6	,	13	,	S	]			
2	f	[	6	,	12	,	S	]			
2	f	[	6	,	11	,	S	]		P1=[6,11]	D1= S
	f	[	6	,	10	,	S	]			
1	f	[	6	,	9	,	S	]			
1	o	[	6	,	9	,	W	]			
1	f	[	5	,	9	,	W	]			
1	a	[	5	,	9	,	S	]			
2	f	[	5	,	8	,	S	]			
2	o	[	5	,	8	,	W	]			
2	f	[	4	,	8	,	W	]			
2	a	[	4	,	8	,	S	]			
3	f	[	4	,	7	,	S	]			
3	o	[	4	,	7	,	W	]			
3	f	[	3	,	7	,	W	]			
3	a	[	3	,	7	,	S	]			
4	f	[	3	,	6	,	S	]			
4	o	[	3	,	6	,	W	]			
4	f	[	2	,	6	,	W	]			
4	a	[	2	,	6	,	S	]			
	f	[	2	,	5	,	S	]		P2=[2,5]	D2= S

Per trovarle più velocemente, è utile osservare che il primo comando r, ad ogni ripetizione sposta in avanti il robot di 2 posizioni e gira verso sinistra, mentre il secondo comando r esegue ad ogni ripetizione un movimento che complessivamente equivale a spostare il robot "in diagonale", ovvero lo sposta di una posizione lungo l'asse delle X e di una lungo l'asse delle Y.

ROBOT con memoria

L' ultimo modello di robot, oltre a eseguire movimenti tramite i comandi **o**, **a** ed **f**, ha una nuova caratteristica: la capacità di memorizzare e richiamare sotto-liste di comandi.
Una sotto-lista è una sequenza di comandi a cui viene attribuito un *numero identificativo*, e che può essere inserita all'interno di un'altra sequenza di comandi proprio utilizzando il *numero identificativo*. In dettaglio, il robot oltre a **o**, **a** ed **f**, esegue due nuovi comandi:

- comando **s**: questo comando è seguito dal *numero identificativo*, poi da una sequenza di comandi chiamata *corpo* e infine dal simbolo **|** che indica la fine del *corpo*
    - un esempio è **s3a,f,o,f,f|**in cui 3 è il *numero identificativo* e **a,f,o,f,f** è il *corpo*
    - l'esecuzione di **s** non provoca alcuno spostamento del robot; accade invece che il robot memorizza al suo interno il *corpo* del comando, come sotto-lista di comandi che viene identificata dal *numero identificativo*
    - attenzione: quando il robot esegue il comando **s**, non esegue i comandi che formano il corpo del comando
- comando **c**: questo comando è seguito da un *numero identificativo*
    - un esempio è **c3** in cui 3 è *numero identificativo*

- quando il robot esegue un comando **c**, se ha eseguito in precedenza un comando **s** che aveva lo stesso numero identificativo del comando **c**, allora il robot esegue tutti i comandi del corpo di tale comando **s**; altrimenti il robot non fa nulla e passa ad eseguire il prossimo comando della lista

Ad esempio, se al robot viene data la lista di comandi [**f,s1f,f,a|f,a,c2,f,c1,f**], il robot si comporta come segue:

1. esegue **f**
2. esegue **s** seguito da **1**, ovvero memorizza al suo interno il corpo **f,f,a** associato al numero identificativo **1**
3. esegue **f** e poi **a** (sono i comandi che vengono subito dopo il simbolo |
4. esegue il comando **c** seguito da **2**: poiché non è stato eseguito in precedenza un comando **s** con numero identificativo pari a **2**, il robot non fa nulla
5. esegue **f** (è il comando che segue **c2**)
6. esegue il comando **c** seguito da **1**: poiché è stato eseguito in precedenza il comando **s** con numero identificativo pari a **1**, il robot esegue i comandi del corpo di tale comando **s**, ovvero **f,f,a**
7. esegue **f** (è il comando che segue **c1**)

Pertanto la lista degli *effettivi spostamenti* di questo robot è [**f,f,a,f,f,f,a,f**]

ESEMPIO 3
Gianluca posiziona il suo robot con memoria sul campo di gara. Lo stato del robot è [11,6,N]. Gianluca attiva il robot con la seguente lista di comandi:
L1 =[s2f,f,o,f|a,c2,f]
Quale sarà lo stato finale S1 del robot, dopo aver eseguito tutti i comandi di L1?
Sposta poi il robot, portandolo nello stato [18,18,E], e lo attiva con la seguente lista di comandi:
L2 =[f,s5f,o,f,a|f,c4,c5,a,c5,f]
Quale sarà lo stato finale S2 del robot, dopo aver eseguito tutti i comandi di L2?
Infine lo sposta nello stato [15,32,S] , e lo attiva con la seguente lista di comandi:
L3 =[s1a,a,f,f|o,c1,a,c1]
Quale sarà lo stato finale S3 del robot, dopo aver eseguito tutti i comandi di L3?

SOLUZIONE

S1	[9,8,N]
S2	[22,19,N]
S3	[17,30,S]

COMMENTI ALLA SOLUZIONE
La lista di comandi L1 inizia con il memorizzare una sotto-lista con numero identificativo 2, poi fa ruotare il robot in senso antiorario e quindi richiama la sotto-lista (che viene eseguita), e infine sposta il robot in avanti di una posizione.
La lista L1 è equivalente alla seguente lista priva di comandi **s** e **c** (compatibile con il modello di robot classico): [**a,f,f,o,f,f**]. Eseguendo tali comandi si trova la risposta al quesito.

Inizio		[	11	,	6	,	N	]
	a	[	11	,	6	,	W	]
	f	[	10	,	6	,	W	]
	f	[	9	,	6	,	W	]
	o	[	9	,	6	,	N	]
	f	[	9	,	7	,	N	]
	f	**[**	**9**	**,**	**8**	**,**	**N**	**]** **S1**

La lista di comandi L2 inizia spostando il robot in avanti. Poi memorizza la sotto-lista [**f,o,f,a**] con numero identificativo 5, e non sposta il robot. Dopo la definizione, c'è un comando che sposta il robot in avanti. Successivamente c'è il comando **c** con numero identificativo 4: poiché in precedenza non è stato memorizzato **s 4**, il robot non si sposta. Successivamente il robot esegue il comando
c5 (la sotto-lista [**f,o,f,a**]). Il comando successivo **a**, ruota il robot in senso antiorario. Poi c'è di nuovo il comando **c5**, e quindi il robot esegue per la seconda volta la sotto-lista [**f,o,f,a**]. Infine, l'ultimo comando sposta in avanti il robot. Quindi la lista L2 è equivalente alla seguente lista di un robot classico [**f,f,f,o,f,a,a,f,o,f,a,f**]. Eseguendo tali comandi si trova la risposta al quesito.

Inizio		[	18	,	18	,	E	]
	f	[	19	,	18	,	E	]
	f	[	20	,	18	,	E	]
	f	[	21	,	18	,	E	]
	o	[	21	,	18	,	S	]
	f	[	21	,	17	,	S	]
	a	[	21	,	17	,	E	]
	a	[	21	,	17	,	N	]
	f	[	21	,	18	,	N	]
	o	[	21	,	18	,	E	]
	f	[	22	,	18	,	E	]
	a	[	22	,	18	,	N	]
	f	[	**22**	**,**	**19**	**,**	**N**	**]** **S2**

La lista di comandi L3 inizia con la memorizzazione della sotto-lista [**a,a,f,f**] con numero identificativo 1. Poi c'è un comando che ruota il robot in senso orario. Successivamente il robot esegue il comando **c1** (esecuzione della sotto-lista [**a,a,f,f**]). Il comando successivo è **a**, che ruota il robot in senso antiorario. Infine, c'è nuovamente il comando **c1** (esecuzione per la seconda volta della sotto-lista [**a,a,f,f**]).
Quindi la lista L3 è equivalente alla seguente lista di un robot classico [**o,a,a,f,f,a,a,a,f,f**]. Eseguendo tali comandi si trova la risposta al quesito.

Inizio		[	15	,	32	,	S	]	
	o	[	15	,	32	,	W	]	
	a	[	15	,	32	,	S	]	
	a	[	15	,	32	,	E	]	
	f	[	16	,	32	,	E	]	
	f	[	17	,	32	,	E	]	
	a	[	17	,	32	,	N	]	
	a	[	17	,	32	,	W	]	
	a	[	17	,	32	,	S	]	
	f	[	17	,	31	,	S	]	
	f	**[**	**17**	**,**	**30**	**,**	**S**	**]**	**S3**

h) SOTTOSEQUENZE

Una sequenza può essere pensata come una lista; per esempio la seguente è una sequenza di numeri interi (non necessariamente distinti):

[15,6,12,18,9,8,10,20,8,4,7]

Una *sottosequenza* è una lista che contiene una parte degli elementi di quella originale, posti nello stesso ordine. Esempi di sottosequenze della lista precedente sono:

[15,18,20,4], [15,6,12,18,7], [9,8,10, 8,4,7]

Non è una sottosequenze della lista precedente la seguente:

[15,18,12,9,8,20,10,4,]

perché gli elementi non compaiono nello stesso ordine di quella data (per esempio 18 e 12 o 20 e 10) .

In certi casi, data una sequenza, è rilevante determinare sottosequenze con certe proprietà; un caso tipico è determinare la sottosequenza più lunga (strettamente) *decrescente*, o quella (strettamente) *crescente*, oppure quella i cui elementi godono di certe proprietà.

N.B. “strettamente” indica che non ci sono elementi ripetuti.

ESEMPIO 1

Un bambino ama guardare i cartoni animati in TV. I genitori gli impongono la regola che dopo aver guardato un programma di durata *d*, può guardare solo programmi di durata minore di *d*. Al bambino viene data la Guida TV del suo canale preferito che contiene l’elenco di tutti i programmi della giornata con le relative durate. Aiutatelo a scegliere il più grande insieme di programmi che può guardare senza infrangere la regola imposta, se la sequenza delle durate dei programmi (espresse in minuti) è la seguente:

[7,20,12,14,13,15,8,4]

SOLUZIONE

[20,14,13,8,4]

COMMENTI ALLA SOLUZIONE

Il problema chiede di determinare la *sottosequenza decrescente più lunga* estratta dalla lista data. Nei casi semplici come questo si può procedere “ad occhio”; nei casi più complessi occorre essere sistematici ed esaminare *tutte* le possibili sequenze.

[7,4]
[20,12,8,4]
[20,14,13,8,4]
[20,15,8,4]

La lista soluzione [20,14,13,8,4] segue immediatamente.

ESEMPIO 2
Si consideri la sequenza di numeri interi:

[8,10,11,12,4,18,6,7,14,10,18,20].

Determinare la più lunga sottosequenza strettamente crescente di numeri pari.

SOLUZIONE

[8,10,12,14,18,20]

COMMENTI ALLA SOLUZIONE
Sottosequenze che iniziano con 8:
[8,10]
[8,10,12]
[8,10,12,18]
[8,10,12,18,20]
[8,10,12,14,18]
[8,10,12,14,18,20]

Le sottosequenze a partire dal 10 o dal 12 non le consideriamo perché sottosequenze delle 6 precedenti. Esaminiamo ancora quelle che iniziano con il 4 con almeno cinque termini:
[4,6,10,18,20]
[4,6,14,18,20]

La soluzione è la lista di lunghezza 6 [8,10,12,14,18,20].

i) FLUSSI IN UNA RETE DI CANALI

Sul fianco di una montagna esistono numerose sorgenti. L'acqua di una sorgente, che si suppone fluire in modo continuo e costante, può scorrere a valle attraverso uno o più canali. Può avvenire che uno o più canali convergano in un punto in cui esiste una sorgente; in tal caso, la loro acqua si aggiunge a quella fornita dalla sorgente raggiunta. Questa situazione è descrivibile con un reticolo di nodi (le sorgenti) collegati da archi (i canali).
La situazione quindi è descritta da due tabelle:

s(<nome della sorgente>,<litri d'acqua erogati al minuto>),

che specifica la quantità d'acqua che sgorga da ogni sorgente (che è un nodo del reticolo),

r(<nome della sorgente a monte>,<nome della sorgente a valle>)

che specifica la presenza di un canale che porta acqua da monte a valle.

Per ogni nodo l'acqua si divide equamente tra i canali che escono (a valle) dal nodo.

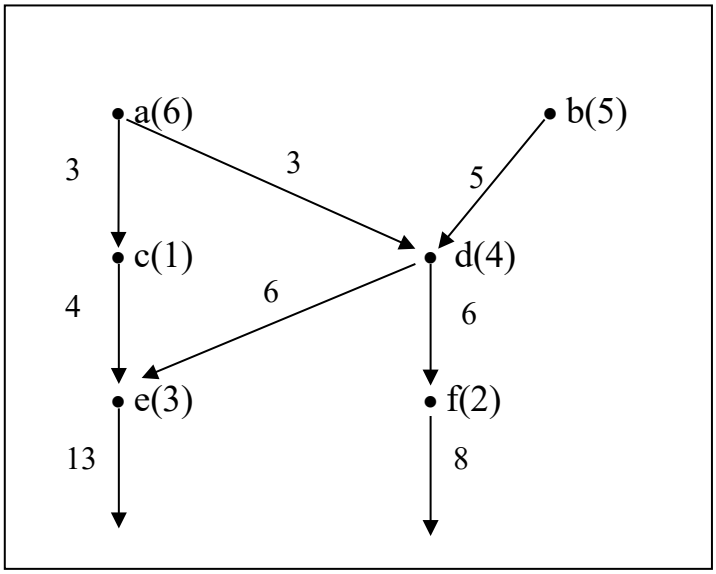
ESEMPIO 1
Un reticolo di canali è descritto dalle seguenti due tabelle:
s(a,6), s(b,5), s(c,1), s(d,4), s(e,3), s(f,2)
r(a,c), r(a,d), r(b,d), r(c,e),r(d,e), r(d,f)
Disegnare il reticolo, evitando incroci, e determinare la quantità di acqua che esce dai nodi c, e, f.

SOLUZIONE

c	4
e	13
f	8

COMMENTI ALLA SOLUZIONE

Occorre essenzialmente disegnare il reticolo; la portata delle sorgenti è assegnata; la soluzione segue applicando le regole per calcolare la portata dei canali. Naturalmente occorre aggiungere dei canali in uscita dai nodi e, f.



Determinare la quantità d'acqua che esce dai nodi c, e, f .

ESEMPIO 2

Un reticolo di canali è descritto dalle seguenti due tabelle:

s(a,3), s(b,3), s(c,4), s(d,6), s(e,6), s(f,6), s(g,6), s(h,4), s(i,6), s(j,7), s(k,5), s(m,9)

r(a,e), r(b,g), r(c,e), r(d,g), r(d,h), r(i,b), r(i,d), r(d,f), r(d,e), r(k,d), r(j,d), r(m,d), r(m,c), r(m,a)

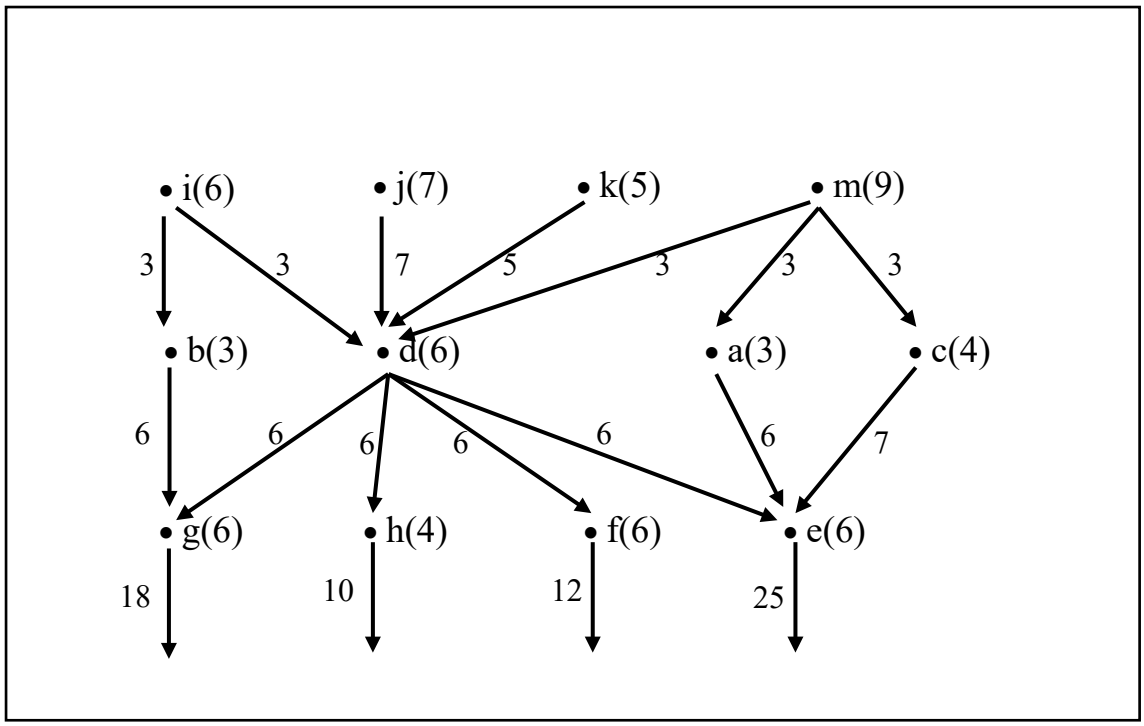
Disegnare il reticolo, evitando incroci, e determinare la quantità di acqua che esce dai nodi e, f, g, h.

SOLUZIONE

e	25
f	12
g	18
h	10

COMMENTI ALLA SOLUZIONE

Occorre essenzialmente disegnare il reticolo; la portata delle sorgenti è assegnata; la soluzione segue applicando le regole per calcolare la portata dei canali. Naturalmente occorre aggiungere dei canali in uscita dai nodi e, f, g, h.



ESEMPIO 3

Una rete di canali è descritta dalle seguenti due tabelle di sorgenti e canali rispettivamente,

s(a,7), s(b,8), s(c,10), s(d,9), s(e,9), s(f,1), s(g,8),
s(h,6), s(i,7), s(j,2), s(k,11), s(l,8), s(m,10), s(n,2);
r(a,e), r(b,e), r(b,f), r(c,f), r(c,g), r(d,g), r(e,h), r(e,i), r(f,i),
r(f,j), r(g,j), r(g,k), r(h,l), r(i,l), r(i,m), r(j,m), r(j,n), r(k,n).

Disegnare la rete, evitando incroci tra i canali, e determinare da quale nodo *finale* esce la quantità maggiore di acqua.

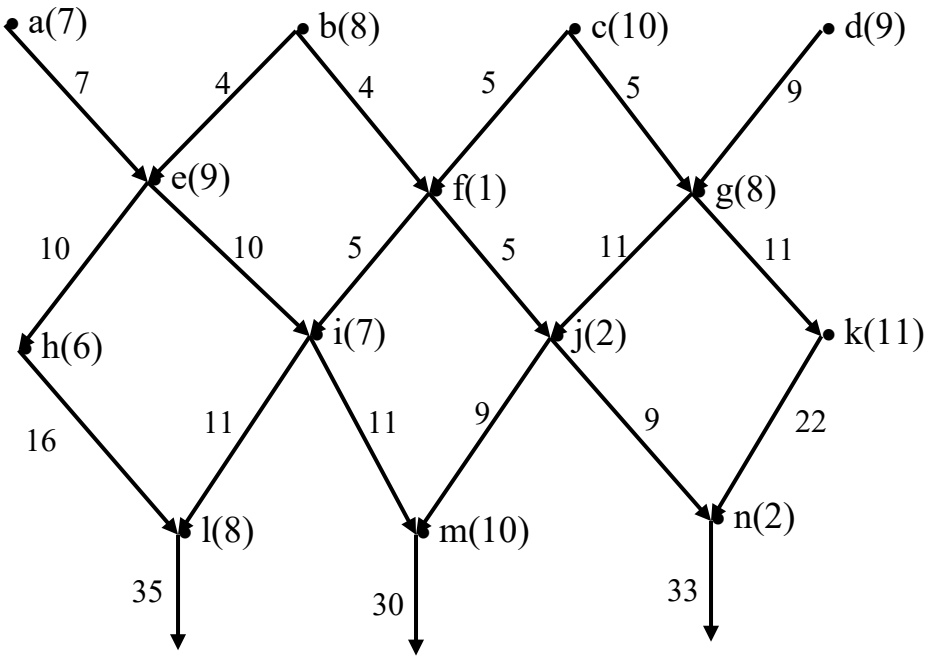
N.B. Un nodo è finale quando non compare come secondo argomento in un termine "r": cioè quando non ha successori (a valle).

SOLUZIONE

nodo finale con maggior portata in uscita	l
portata in uscita (verso valle) di tale nodo	35

COMMENTI ALLA SOLUZIONE

Occorre essenzialmente disegnare il reticolo; nel disegno ogni sorgente è rappresentata da un nodo (punto) con nome e portata assegnata; ogni canale è rappresentato da un segmento orientato verso valle ed è etichettato con la portata calcolata. La soluzione si ottiene, appunto, applicando le regole per calcolare la portata dei canali. Naturalmente occorre aggiungere i canali in uscita dai nodi l, m, n.


a(7)
b(8)
c(10)
d(9)
7
4
4
5
5
9
e(9)
f(1)
g(8)
10
10
5
5
11
11
h(6)
i(7)
j(2)
k(11)
16
11
11
9
9
22
l(8)
m(10)
n(2)
35
30
33

4. ELEMENTI DI PSEUDOLINGUAGGIO

4.0 CHE COSA SAPPIAMO .....

Ormai da diversi anni esistono congegni elettronici chiamati computer che sono in grado di aiutare gli esseri umani a risolvere problemi.
Uno o più esseri umani :

pensano ad un problema
lo descrivono in un testo, utilizzando una lingua parlata (linguaggio naturale) e inserendo magari delle formule.
Per trasferire il problema al computer "traducono il testo" in una *procedura* utilizzando termini e simboli dello *pseudolinguaggio*.

Questa procedura sarà poi inserita in un computer utilizzando un "linguaggio di programmazione" o qualche "applicativo".
Finalmente il computer , eseguendo tale procedura ,fornirà possibili risposte al problema.

ESEMPIO
Scrittura problema in linguaggio naturale

Problema
Il lato di un quadrato misura 6 m. Calcolare il perimetro P e l'area A di tale quadrato.
Dato del problema : lato del quadrato L = 6 m
Formule risolventi : perimetro quadrato P = L*4
area quadrato A = L*L

Traduzione problema in una procedura utilizzando lo pseudolinguaggio

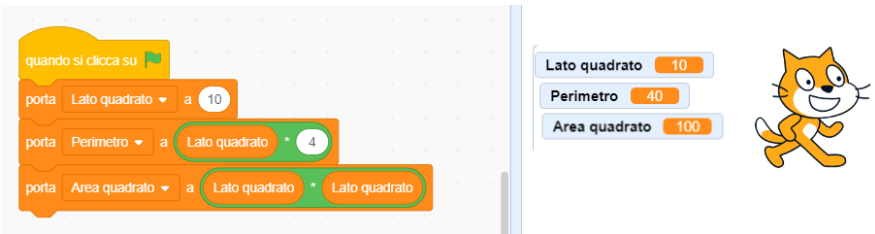
```
procedure QUADRATO;
variables  L, P, A  integers;
read L ;
P = L * 4;
A = L * L;
write P, A;
end procedure;
```

Riscrittura della procedura QUADRATO (su un computer) utilizzando:

Il programma (applicativo) *foglio elettronico* :

	A	B	C	D	E
1					
2	lato del quadrato	L	10	m	
3					
4	perimetro del quadrato	P	=C2*4	m	
5	area del quadrato	A	=C2*C2	mquadri	
6					

Il linguaggio di programmazione scratch:



4.1 LETTURA DI UNA PROCEDURA E SIMULAZIONE DELLA UNA SUA ESECUZIONE.

a)LE VARIABILI

Iniziamo con una metafora.In una cassettiera ci sono dei cassetti individuati dalle lettere A, B, C, D, E, F. In ciascun cassetto ci *può* essere un solo foglio su cui è scritto un numero *intero*. La scrittura

C = A+B;

significa: "*sommare i numeri scritti sui fogli dei cassetti A e B, scrivere il risultato su un nuovo foglio e inserire questo foglio nel cassetto C, dopo aver eliminato il foglio (eventualmente) presente in C*". Se all'inizio nei fogli di A e B sono scritti rispettivamente i numeri 12 e 7, a operazione eseguita in C si trova un foglio su cui è scritto 19.

Così, la scrittura:

D = C+A-B;

significa:

sommare i numeri scritti sui fogli dei cassetti C e A,
sottrarre alla somma il numero scritto sul foglio del cassetto B,
scrivere il risultato su un nuovo foglio e inserire questo foglio nel cassetto D dopo aver eliminato il foglio (eventualmente) presente in D".

In questo caso se all'inizio nei fogli di A,B e C sono scritti rispettivamente i numeri 12,7 e 10, a operazione eseguita in D si trova un foglio su cui è scritto 15 (12+10-7).

N.B. Per brevità diciamo "*il numero contenuto in C*" invece di "*il numero scritto sul foglio contenuto in C*".

Invece di parlare di cassetti e di numeri scritti su fogli, si può ricorrere a una descrizione, più astratta.
Le lettere maiuscole A, B, C, … sono chiamate "*variabili*" (invece di cassetti) e i numeri sui fogli sono detti "*valori*" di quelle variabili.

METAFORA	ASTRAZIONE
cassetto A	variabile A
foglio in A	contenuto della variabile A
numero scritto sul foglio	valore della variabile A

In una procedura le variabili utilizzate vengono dichiarate all'inizio e si specifica quali sono i valori che possono accettare (per ora solo numeri interi).
Ad esempio se A, B, C sono tre variabili, in pseudolinguaggio si scrive :

variables A, B, C integer;

Dovendo assegnare alla variabile A un valore dall'esterno scriviamo:

read A ;

Volendo sapere quale valore contiene la variabile B, scriviamo:

write A ;

N.B. Si usa spesso dire che l'istruzione read assegna i valori in *input*, mentre l'istruzione write mostra i valori in *output.*

b) LEGGERE UNA PROCEDURA E SIMULARE UNA SUA ESECUZIONE

Prendendo come esempio la procedura QUADRATO abbiamo:

La prima riga indica che inizia la procedura a cui è stato dato un NOME (QUADRATO)

procedure QUADRATO;

Nella seconda si dichiarano le variabili utilizzate e specificato il loro tipo (integer)
variables L, P, A integer ;

Nella terza riga si assegna alla variabile L un valore numerico (misura del lato)
read L;

Seguono due righe in cui si calcola il perimetro P e l'area A; i valori numerici calcolati sono assegnati alle variabili P, A.
Nella penultima riga si chiede di vedere (o stampare) i valori contenuti in P e A
write P, A;

L'ultima riga indica il termine della procedura
end procedure;

N.B. Ogni riga della procedura si dice *statement* (o *istruzione*) e termina sempre con ";".

Un secondo esempio è dato dalla procedura CALCOLI

procedura	Lettura
procedure CALCOLI; variables A,B,C,D integer; read A,B,C; D = A+B; A = D+C; B = A*B; write A, B, D; end procedure;	la procedura ha nome CALCOLI ci sono quattro variabili A,B,C,D di tipo numerico intero alle variabili A,B,C sono assegnati valori numerici in D si "scrive" (memorizza) la somma di A e B in A si memorizza la somma di D e C in B si memorizza il prodotto tra A e B si stampano i valori attuali di A,B,D la procedura è terminata

Simulare l'esecuzione di una procedura significa creare una tabella in cui vengono dati valori numeri a certe variabili (istruzione read) e si svolgono tutte le successive azioni fino al write compreso.

Nel caso della procedura QUADRATO in cui a L diamo il valore 14 abbiamo:

	L	P	A
read A	10		
P= L*4	10	40	
A= L*L	10	40	100
write		40	100

In CALCOLI quando alle variabili A,B e C vengono assegnati rispettivamente i valori 5, 8 e 3.

	A	B	C	D
read A, B, C	5	8	3	
D= A + B	5	8	3	13
A= D + C	16	8	3	13
B= A * B	16	128	3	13
write A,B,D	16	128		13

N.B. In questa procedura le variabili A e B hanno alla fine valori diversi da quelli iniziali. Ricordando la meta fora delle variabili, è successo che in questi cassetti sono stati tolti i fogli iniziali e introdotti altri fogli con scritti numeri derivati da calcoli successivi.

Per finire due esempi di esercizi di gara.

A) Si consideri la seguente procedura PROVA1.

```
procedure PROVA1;
variables A, B, C, D integer;
read  A, B;
C = A + B;
D =A * B;
A =C+B;
B = (A+B) *(A- B);
write C, D, A, B;
end procedure;
```

I valori in input sono: 4 per A, 2 per B; determinare i valori di output di A, B, C, D e scriverli nella seguente tabella.

A	
B	
C	
D	

SOLUZIONE

A	8
B	60
C	6
D	8

COMMENTI ALLA SOLUZIONE
Il problema si risolve compilando la tabella di esecuzione della procedura.

	A	B	C	D
read A,B	4	2		
C=A+B	4	2	4+2=6	
D=A*B	4	2	6	4*2=8
A=C+B	6+2=8	2	6	8
B=(A+B)*(A-B)		(8+2)*(8-2)=60		
write A,B,C,D	8	60	6	8

Notare che i valori di certe variabili cambiano più volte: per esempio le variabili A e B acquisiscono un primo valore in input e successivamente cambiano valore con le ultime due operazioni.

B) Si consideri la seguente procedura PROVA2 (scritta in maniera sintatticamente scorretta: il simbolo X non è definito).

```
procedure PROVA2;
variables A, B, C, D integer;
D =0;
read A, B, C;
D =  A + B + C + X;
write  D;
end  procedure;
```

Trovare, tra le variabili dichiarate nella procedura, il nome da sostituire a X per ottenere in output 21 per D se i valori in input sono 2 per A, 5 per B e 7 per C.

nome della variabile da sostituire a X	

SOLUZIONE

nome della variabile da sostituire a X	C

COMMENTI ALLA SOLUZIONE
Poiché A, B, C valgono rispettivamente 2, 5 e 7, occorre che il risultato di

A + B + C + X

cioè

2 + 5 + 7 + X

valga 21; questo richiede che il valore di X sia 7, cioè quello di C.

Una altra maniera di procedere alla soluzione è di esaminare tutti i possibili casi: poiché sono dichiarate quattro variabili: A, B, C, D, allora si possono esaminare i risultati delle quattro possibili sostituzioni:

l'espressione A + B + C + A vale 2 + 5 + 7 + 2 = 16
l'espressione A + B + C + B vale 2 + 5 + 7 + 5 = 19
l'espressione A + B + C + C vale 2 + 5 + 7 + 7 = 21
l'espressione A + B + C + D vale 2 + 5 + 7 + 0 = 14

La soluzione segue immediatamente.

4.2 L'ALTERNATIVA "if"

Ricordiamo che i simboli "<" e ">" significano rispettivamente *minore di* e *maggiore di.*
Allora leggiamo l'affermazione (*predicato*) 2 < 3 come "il numero 2 è minore del numero 3"
e diciamo che il predicato è *vero*
Mentre il predicato 5 > 7 afferma che "5 è maggiore di 7" ed è *falso.*

Ma se A e B sono variabili integer in una procedura cosa significa che A < B ?

Per rispondere alla domanda bisogna:
sapere quali sono i valori numerici di A e B nel momento in cui esamina il predicato
se per situazioni precedenti si ha A= 5 e B= 7 allora il predicato sarà vero
se per situazioni precedenti si ha A= 7 e B= 5 allora il predicato sarà falso

In una procedura spesso si pone una "alternativa" decisa dal valore di un *predicato*: se il predicato è *vero* si fanno alcune cose, se è *falso* se ne fanno altre.
In pseudolinguaggio la situazione in esame viene così descritta :

```
variables A, B, M integer;
…
if A > B
                then M = A;
                else  M = B;
endif;
write M;
…
```

dove if ….. then …. else ha il seguente significato:
controllare il valore di verità (vero, falso) del predicato A < B
se è vero allora (then) la variabile M prende il valore numerico che c'è in A
altrimenti (else) la variabile M prende il valore numerico che c'è in B perché il predicato è falso

Esempio.
Situazione 1 A=2 e B=5 segue A < B (2 < 5) vero per cui si esegue il then e M = 2 ;
situazione 2 A=7 e B=3 segue A < B (7 < 3) falso per cui si esegue else e M = 3 ;
situazione 3 A=B=5 segue A < B (5 < 5) falso per cui si esegue else e M = 5 ;

Naturalmente al posto di "A > B" si possono usare altri predicati:
"A = B" A uguale a B
"A < B" A minore di B
"A <= B" A minore o uguale a B
"A >= B" A maggiore o uguale a B

Quando *manca* il ramo "else" (cioè quando occorre fare alcune cose se il predicato è *vero*, ma non si deve fare nulla se è *falso*), si usa la forma abbreviata :

```
variables A, B, M integer;
…
M = B;
if A > B then M = A; endif;
write M;
…
```

La presenza di **parentesi graffe** che racchiudono una sequenza di istruzioni sta ad indicare che tutte le istruzioni specificate all'interno delle parentesi devono essere eseguite in sequenza. Se in una procedura compaiono le parentesi graffe all'interno di una alternativa semplice, allora tutta la sequenza di istruzioni specificate all'interno dovranno essere eseguite sulla base del verificarsi della condizione espressa nell'alternativa.

```
variables A, B, C integer;
read A, B;
if  B > A then {
  C = B;
  B = A;
  A = C;
} ; endif;
write A,B;
end procedure;
```

Le tre istruzioni specificate verranno eseguite se e solo se B è maggiore di A.
Caso 1

	v/f	A	B	C
read A, B		3	5	
if B > A	v	3	5	
C= B		3	5	5
B= A		3	3	5
A= C		5	3	5
write A,B		5	5	

Caso 2

	v/f	A	B	C
read A, B		6	4	
if B > A	f	6	4	
C= B non eseguita		6	4	
B= A non eseguita		6	4	
A= C non eseguita		6	4	
write A,B		6	4	

Esempio completo di procedura con "if"

procedura	lettura

procedure BETA;
variables A, B, C, D, F integer;
read A, B, C;
D = 0;
if C > A then F = A + B;
else F = B + C;
endif;
if C > F then F = C;
else D = C;
endif;
write D, F;
end procedure;
	il nome della procedura è BETA
	le variabili sono A,B,C,D,F possono contenere solo numeri interi
	alle variabili A,B,C vengono assegnati numeri interi
	nella variabile D viene memorizzato il numero 0
	se C > A è vero esegui then F = A + B
	se C > A è falso esegui else F = B + C
	termina il primo if
	se C > F è vero esegui then F = C
	se C > F è falso esegui else D = C
	termina il secondo if
	stampa i valori di D , F
	termine della procedura

I valori in input sono: 125 per A, 125 per B e 113 per C; determinare i valori di output.
SOLUZIONE

D	113
F	238

COMMENTI ALLA SOLUZIONE
La soluzione segue immediatamente eseguendo passo passo la procedura; i valori delle variabili sono mostrati nella seguente tabella.

	v/f	A	B	C	D	F
read A,B,C		125	125	113		
D = 0		125	125	113	0	

C > A	f	125	125	113	0	125+113=238
C > F	f	125	125	113	113	238
write D,F					113	238

4.3 LA RIPETIZIONE.

a)Il ciclo "for"

In molti problemi, la soluzione si ottiene ripetendo le medesime operazioni un certo numero di volte. Un esempio preso dalla geometria è il seguente:

Problema. Calcolare perimetro e area di quattro quadrati aventi rispettivamente i lati lunghi 3,4,6,10 metri.

Soluzione con carta e penna.
Dobbiamo ripetere 4 volte quanto visto nel problema del quadrato nel paragrafo 1

Dato del problema : lato del quadrato L = 3 m
perimetro quadrato P = L*4 = 3*4 = 12 m
area quadrato A = L*L = 3*3 = 9 mq
Dato del problema : lato del quadrato L = 4 m
perimetro quadrato P = L*4 = 4*4 = 16 m
area quadrato A = L*L = 4*4 = 16 mq
Dato del problema : lato del quadrato L = 6 m
perimetro quadrato P = L*4 = 6*4 = 24 m
area quadrato A = L*L = 6*6 = 36 mq
Dato del problema : lato del quadrato L = 10 m
perimetro quadrato P = L*4 = 10*4 = 40 m
area quadrato A = L*L = 10*10 = 100 mq

La procedura QUATTRO-QUADRATI risolve il problema utilizzando per 4 volte all'interno di un "ciclo for" la sequenza di istruzioni

```
read L;
P = L * 4;
A= L * L;
write P,A;
```

La *sintassi* del ciclo for in pseudolinguaggio è la seguente:

```
 for K from 1 to 4 step 1 do;
    <istruzioni da ripetere>
 endfor;
```

da leggere "per K da 1 a 4 con passo 1 esegui <istruzioni da ripetere>" .
In modo più esplicito "ripetere 4 volte le istruzioni <istruzioni da ripetere> assegnando alla variabile K i valori 1,2,3,4".

N.B. Con step 1 si intende "ad ogni ciclo aumenta di uno il valore di K"
Quindi se K = 1 si ha successivamente K =1+1=2,3,4

```
procedure QUATTRO-QUADRATI
variables L, P, A, K integer;
for K from 1 to 4 step1 do;
    read L;
    P = L * 4;
    A= L * L;
    write P,A;
end for;
end procedure;
```

Tabella di esecuzione con L che assume i valori 3,4,6 e 10.

	K	L	P	A
	1			
read L		3		
P = L*4		3	12	
A = L*L		3	12	9
write P,A				
	2			
read L		4		
P = L*4		4	16	
A = L*L		4	16	16
write P,A				
	3			
read L		6		
P = L*4		6	24	
A = L*L		6	24	36
write P,A			24	36
	4			
read L		10		
P = L*4		10	40	
A = L*L		10	40	100
write P,A			40	100

Altri esempi completi.

```
procedure ESEMPIO1;
variables A, B, K integer;
A=0;
B=0;
for K from 1 to 4 step 1 do;
    A = A + 1;
    B = B + K;
endfor;
write A, B;
end procedure;
```

tabella di esecuzione

	K	A	B
A=0		0	
B=0		0	0
	1		
A=A+1	1	1	
B=B+K	1	1	1
	1+1=2		
A=A+1	2	1+1=2	1
B=B+K	2	2	1+2=3
	2+1=3		
A=A+1	3	3	3
B=B+K	3	3	3+3=6
	3+1=4		
A=A+1	4	3+1=4	6
B=B+K	4	4	6+4=10
Write A, B		4	10

```
procedure ESEMPIO2;
variables A, B, K integer;
A = 0;
B =0;
for K from 1 to 10 step 1 do;
    A = A+K;
    B = B+K*K;
endfor;
write A, B;
end procedure;
```

tabella di esecuzione

	K	A	B
A=0		0	
B=0		0	0
	1		
A=A+K	1	0+1=1	
B=B+K*K	1	1	0+1*1=1
	1+1=2		
A=A+K	2	1+2=3	1
B=B+K*K	2	3	1+2*2=5
	2+1=3		
A=A+K	3	3+3=6	5
B=B+K*K	3	6	5+3*3=14
	3+1=4		
A=A+K	4	6+4=10	14
B=B+K*K	4	10	14+4*4=30
write A, B		10	30

```
procedure ESEMPIO 3;
variables A, B, M, N, K integer;
        read A;
        M =0;
        N =0;
        for K from 1 to 10 step 1 do;
                read B;
                if A > B        then M = M + A; endif;
                if A < B        then  N = N + A;  endif;
        endfor;
        write M, N;
        end procedure;
```

I valori di input per A è 5 e per B sono rispettivamente: 9, 3, 7, 2, 8, 5, 1, 4, 4, 5. Determinare i valori di output.
SOLUZIONE

M	25
N	15

COMMENTI ALLA SOLUZIONE
Basta eseguire, passo a passo, le operazioni indicate: in N va la somma di tante volte il valore di A quante volte questo è più piccolo di (quello di) B (5 volte); in M va la somma di tante volte il valore di A quante volte questo è più piccolo di (quello di) B (3 volte).

	v/f	K	A	B	M	N
read A			5			
M=0			5		0	
N=0			5		0	0
		1			0	0
read B		1	5	9	0	0
if A > B	f	1	5	9	0	0
if A < B	v	1	5	9	0	0+5=5
		2				
read B		2	5	3	0	5
if A > B	v	2	5	3	0+5=5	5
if A < B	f	2	5	3	5	5
		3				
read B		3	5	7	5	5
if A > B	f	3	5	7	5	5
if A < B	v	3	5	7	5	5+5=10
		4				
read B		4	5	2		
if A > B	v	4	5	2	5+5=10	10
if A < B	f	4	5	2	10	10
		5				
read B		5	5	8	10	10
if A > B	F	5	5	8	10	10
if A < B	v	5	5	8	10	10+5=15

		6				
read B		6	5	5	10	15
if A > B	f	6	5	5	10	15
if A < B	f	6	5	5	10	15
		7				
read B		7	5	1	10	15
if A > B	v	7	5	1	10+5=15	15
if A < B	f	7	5	1	15	15
		8				
read B		8	5	4	15	15
if A > B	v	8	5	4	15+5=20	15
if A < B	f	8	5	4	20	10+5=15
		9				
read B		9	5	4	15	15
if A > B	v	9	5	4	20+5=25	15
if A < B	f	9	5	4	25	15
		10				
read B		10	5	5	25	15
if A > B	f	10	5	5	25	15
if A < B	f	10	5	5	25	15
write M,N					25	15

b) Il ciclo "while"

Quando il numero di ripetizioni di un gruppo di azioni non può essere precisato ma dipende dal verificarsi di un certo predicato utilizziamo la struttura "while" in alternativa al for.
Essa ripete certe azioni *"mentre è vera una certa condizione*".

La sua sintassi è la seguente:

```
while  <condizione vera> do;
   <istruzioni da eseguire>
endwhile;
```

```
procedure ESEMPIO1;
variables A, B, K integer;
A= 0;
B=10;
K=0;
while A<B do;
      K = K+1;
      A = K*K+A;
endwhile;
write A;
end procedure;
```

N.B. il ciclo while viene eseguito mentre è vero il predicato A < B. All'interno del ciclo A aumenta il suo valore mentre B rimane fisso. Quando avremo A < B falso si passerà direttamente all'istruzione successiva a "endwhile" che nel nostro casa è write A.

tabella di esecuzione

	v/f	K	A	B
A=0			0	
B=10			0	10
K =0		0	0	10
A < B	v			
K=K+1		1	0	10
A=K*K+A		1	1*1+0=1	10
A < B	v			
K=K+1		1+1=2	1	10
A=K*K+A		2	2*2+1=5	10
A < B	v			
K=K+1		2+1=3	5	10
A=K*K+A		3	3*3+5=14	10
A < B	f			
write A			14	

Dopo la terza iterazione il valore di A non è più minore di quello di B e il ciclo si arresta; A quindi vale 14.

4.4 UTILIZZO DELLE VARIABILI REAL

Oltre a valori interi le variabili possono contenere valori razionali, cioè numeri "con la virgola": in questo contesto, però, si userà sempre il "." come separatore decimale. Le variabili di questo tipo si dicono "float". Corrispondentemente alle variabili di tipo float si usano le costanti di tipo float: si scrivono col punto decimale seguito da almeno una cifra, come in

5.45
5.0
45362.9877

Il seguente è un esempio di procedura che usa variabili float.

procedura	lettura
procedure NREALE;	
variables A, B, C integer;	dichiarazione di variabili intere
variables TF, SF, R float;	dichiarazione di variabili razionali
B =2;	assegnazione della costante intera B
A =6;	assegnazione della costante intera A
TF =2.0;	assegnazione della costante razionale TF
SF =5.0;	assegnazione della costante razionale SF
C =A/B;	calcolo e assegnazione alla variabile intera C
R = SF/TF;	calcolo e assegnazione alla variabile float R
write C, R;	risulta C =3 e R =2.5
end procedure;	

Si ricorda che le normali operazioni aritmetiche sono indicate con +, −, *, /. Spesso è usato anche l'elevamento a potenza, col simbolo ^. Per esempio 2^3 = 8; A^B: se A ha valore 2 e B ha valore 3, il risultato dell'espressione è 8; se A ha valore 2.0 e B ha valore 3, il risultato dell'espressione è 8.0.

4.5 UTILIZZO DELLE VARIABILI STRING

Con il termine *stringa* si intende una generica sequenza di caratteri sia alfabetici che numerici o altro. Sono esempi di stringhe le seguenti sequenze:

012

3A5B
Alberto@

In una procedura ci possono essere variabili che contengono stringhe, e per questo si dicono di tipo "*string*". Una costante string viene scritta racchiusa tra apici:

'alpha'
'Giuseppe'
'1200'
''
' '

N.B. La (costante) stringa '1200' *non* è il numero intero 1200: in particolare non si può sommare o sottrarre; la costante '' è la stringa *vuota*; la costante ' ' è la stringa *spazio*.

Se A è una variabile di tipo string, con la scrittura *len(A)* indichiamo la lunghezza della stringa memorizzata in A, ossia il numero dei caratteri che formano la stringa.
Esempio. A = 'alpha' len(A)=5
B ='un cane nero' len(B)= 12 (anche i due spazi sono "caratteri" della stringa)

Se A è una variabile di tipo string, con la scrittura A(n1,n2) indichiamo la stringa formata dai caratteri di A che ci sono a partire dal termine nella posizione n1 fino al termine nella posizione n2 compresa.
Esempio. M='maggio'
M(1,1)='m' M(1,3)='mag' M(3,5)='ggi'

Se N e C sono due variabili string, con la scrittura N&C (leggere "N unita a C" o "N concatenata a C") indichiamo una nuova stringa formata dalla stringa di N a cui uniamo alla sua destra la stringa di C
Esempio N= 'Rossi ' C= 'Alberto'
N&C = 'Rossi Alberto'

I seguenti sono esempi di procedure che usano variabili string e integer.

procedura	lettura
procedure STR1;	la procedura ha nome STR1
variables L, integer;	dichiarazione di variabile intera
variables AS, BS, CS, DS string;	dichiarazione di variabili stringhe
BS = 'Aprile';	assegnazione della costante string 'Aprile'
L = len(BS);	calcolo della lunghezza della stringa BS;
AS = 'mese di ' & BS;	assegnazione a AS del valore stringa ottenuto concatenando una costante e (il valore di) una variabile: AS vale 'mese di Aprile';
CS = BS(1,3);	assegnazione a CS del valore stringa ottenuto con l'operazione BS(1,3), che estrae da BS la sottostringa che va dal carattere 1 (primo) al carattere 3 (terzo); il valore di CS è 'Apr'
DS = BS(4,L);	assegnazione a DS del valore stringa ottenuto con l'operazione BS(4,L), che estrae da BS la sottostringa che va dal carattere 4 (quarto) al carattere L (ultimo); il valore di DS è 'ile'
end procedure;	fine procedura

procedura	lettura
procedure STR2;	la procedura ha nome STR2
variables L, J integer;	dichiarazione di variabili intere
variables AS, BS, string;	dichiarazione di variabili stringhe
BS =‘Aprile’;	assegnazione della costante string ‘Aprile’
L = len(BS);	calcola la lunghezza del valore (stringa) di BS che è 6
for J from 1 to L step 1 do;	ripetere L volte
AS = BS(J,J) ;	l’operazione BS(J,J) estrae dalla stringa che BS ogni volta il carattere che si trova in posizione J a partire da sinistra e lo assegna ad AS;
write AS;	La stampa di AS (di volta in volta) sarà: ‘A’, ‘p’, ‘r’, ‘i’, ‘l’, ‘e’
endfor;	fine del for
end procedure;	fine procedura

Seguono due esempi di problemi su stringhe che riguardano esercizi di pseudolinguaggio nelle gare.

PROBLEMA1
Si consideri la seguente procedura STR3.

```
procedure PROVA4;
variables J, L, C integer;
variables A, B string;
read A;
C = 0;
L = len(A);
for J from 1, step 1 to L do;
        B = A(J,J) ;
        if B = ‘a’        then C = C + 1; endif;
endfor;
write C;
end procedure;
```

Il valore di input per A è: ‘9, a, b, 2, ac, 5, a, 4, 4a, 5b’, Determinare il valore di output per C e scriverlo nella casella sottostante.

C	

SOLUZIONE

C	4

COMMENTI ALLA SOLUZIONE
La procedura inizialmente calcola la lunghezza della costante stringa assegnata ad A; ci sono 31 caratteri poiché bisogna contare numeri, lettere, virgole e spazi.
Successivamente inizia un ciclo for da 1 a 31 (=len(A)) che esamina ogni simbolo contenuto in A.
Il contatore C si incrementa ogni volta che la sottostringa B di A contiene il simbolo (lettera) ‘a’.
Pertanto, come si verifica facilmente, C vale 4.

PROBLEMA 2
Premessa.
Sia S=’tennis’ e C=’r’
L'operatore ==, verifica se due stringhe sono uguali. Dunque saranno **veri** i seguenti:
'a' == 'a'
S == 'tennis'
'r' == C
e **falsi** i seguenti
'a' == 'b'
'Tennis' == S (diverse per via della maiuscola)
C == 'C' (poiché confronto ‘C’ con 'r')

Consideriamo la procedura seguente:

```
procedure STR4;
variables I, L, C integer;
variables CAR, FRASE string;
read FRASE, CAR;
L = len(FRASE);
C = 0;
for I from 1 to L step 1 do;
   if FRASE(I,I) == CAR then C = C + 1; endif;
 endfor;
write C;
end procedure;
```

Sapendo che il valore in input per FRASE è 'Nel mezzo del cammin di nostra vita' e per CAR è il carattere 'm', calcolare il valore di output per C.

C	

SOLUZIONE

C	3

COMMENTI ALLA SOLUZIONE
La procedura conta quante volte il carattere CAR compare nella stringa FRASE. In questo caso 'm' compare 3 volte.
L indica la lunghezza della frase, in questo caso 35 (29 lettere a cui dobbiamo aggiungere 6 spazi tra le parole).
I assumerà tutti i valori da 1 a 35, e FRASE(I,I) conterrà, nell'ordine, tutti i caratteri della frase: 'N', poi 'e', poi 'l', poi ' ', poi 'm' e così via. Ogni volta che FRASE(I,I) sarà uguale a CAR, cioè a 'm', il contatore C verrà incrementato di 1.