ESERCIZIO 1
Si faccia riferimento all'Allegato A - OPS 2016, problema ricorrente REGOLE E DEDUZIONI, pagina 2.

PROBLEMA
Sono date le seguenti regole:

regola(1,[n,t,g],y) regola(2,[n],k) regola(3,[n],g)
regola(4,[g,n,y],f) regola(5,[q],n) regola(6,[p],n)
regola(7,[t,g],n) regola(8,[n],r) regola(9,[p,n,g],m)
regola(10,[t],g) regola(11,[g,m,n],d) regola(12,[r,k],e)

Trovare:
1. la lista L1 che descrive il procedimento per dedurre **d** conoscendo [**p**];
2. la lista L2 che descrive il procedimento per dedurre **e** conoscendo [**q**];
3. la lista L3 che descrive il procedimento per dedurre **f** conoscendo [**t**].

L1	[ ]
L2	[ ]
L3	[ ]

SOLUZIONE

L1	[6,3,9,11]
L2	[5,2,8,12]
L3	[10,7,1,4]

COMMENTI ALLA SOLUZIONE
Per la prima domanda, si osservi che **d** è deducibile con la regola 11 che ha come antecedenti **g**, **m** ed **n** (tutti incogniti). D'altra parte, a partire dal dato **p**, è possibile applicare solamente la regola 6, deducendo **n**, conoscendo il quale si può applicare la regola 2, per dedurre **k**, o la regola 3 per dedurre **g**, o la regola 8 per dedurre **r**; è chiaro che conviene applicare la regola 3 perché, noti **p**, **n** e **g**, si può usare la regola 9 per dedurre **m** e quindi la regola 11, come visto sopra, per dedurre **d**. Il procedimento è quindi [6,3,9,11].

Per la seconda domanda, **e** è deducibile con la regola 12 che ha come antecedenti **r** e **k** (entrambi incogniti. A partire dal dato **q**, è possibile applicare solo la regola 5, deducendo **n**, conoscendo il quale si può applicare la regola 2, per dedurre **k**, o la regola 3 per dedurre **g**, o la regola 8 per dedurre **r**: stavolta è chiaro che conviene applicare la regola 2 e la regola 8 perché **k** ed **r** sono antecedenti della regola 12. Il procedimento è quindi [5,2,8,12].

Per la terza domanda, **f** è deducibile solo con la regola 4, che ha come antecedenti **g**, **n** e **y** che sono incogniti. Conviene ancora cercare le regole applicabili a partire dai soli dati: in questo caso la regola 10, che, da **t**, permette di dedurre **g**; successivamente, noti **t** e **g**, con la regola 7 si deduce anche **n**. A questo puntosi può applicare la regola 1 e dedurre **y**: questo consente di applicare la regola 4. Il procedimento è quindi [10,7,1,4].

ESERCIZIO 2
Si faccia riferimento all'Allegato A - OPS 2016, problema ricorrente PERCORSI IN UN GRAFO, pagina 6.

PROBLEMA
È dato un grafo descritto dal seguente elenco di archi:

arco(n6,n2,3)	arco(n3,n8,2)	arco(n1,n5,3)
arco(n6,n3,6)	arco(n5,n4,1)	arco(n7,n2,6)
arco(n4,n8,8)	arco(n1,n7,1)	arco(n2,n4,2)
arco(n4,n7,4)	arco(n6,n8,3)	

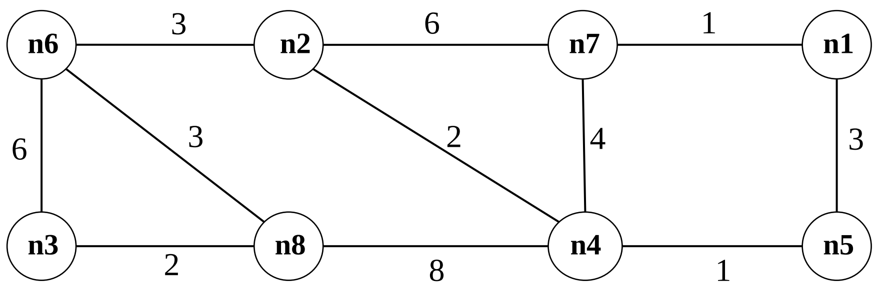
Disegnare il grafo e trovare:
- la lista L1 del percorso più breve tra n1 e n3 che passa per il nodo n6;
- la lista L2 del percorso più lungo (senza passare più volte per uno stesso nodo) tra n1 e n3;
- il numero N di percorsi diversi da n1 a n3 che passano per tutti i nodi.

L1	[ ]
L2	[ ]
N	

SOLUZIONE

L1	[n1, n5, n4, n2, n6, n8, n3]
L2	[n1, n7, n2, n4, n8, n6, n3]
N	1

COMMENTI ALLA SOLUZIONE
Un possibile *layout* del grafo è mostrato nella seguente figura.



N.B. Il grafo è planare, cioè è possibile disegnarlo su un piano senza che gli archi si incrocino. Le lunghezze associate agli archi (che, per esempio, rappresentano delle strade) non sono legate alle reali lunghezze dei segmenti che li rappresentano nel grafo.

I percorsi semplici tra n1 e n3 sono i 14 elencati di seguito.

PERCORSO	LUNGHEZZA	
[n1, n5, n4, n8, n3]	14	lunghezza minore, non passa per n6
[n1, n5, n4, n8, n6, n3]	21	
[n1, n5, n4, n7, n2, n6, n3]	23	
[n1, n5, n4, n7, n2, n6, n8, n3]	22	passa per tutti i nodi
[n1, n5, n4, n2, n6, n3]	15	
[n1, n5, n4, n2, n6, n8, n3]	14	lunghezza minore, passa per n6

[n1, n7, n2, n4, n8, n3]	19	
[n1, n7, n2, n4, n8, n6, n3]	26	lunghezza maggiore
[n1, n7, n2, n6, n3]	16	
[n1, n7, n2, n6, n8, n3]	15	
[n1, n7, n4, n8, n3]	15	
[n1, n7, n4, n8, n6, n3]	22	
[n1, n7, n4, n2, n6, n3]	16	
[n1, n7, n4, n2, n6, n8, n3]	15	

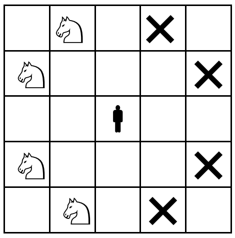
N.B. Questi percorsi possono essere facilmente "organizzati" in un albero; la radice è il nodo di partenza, n1; ogni nodo dell'albero ha tanti figli quanti sono i nodi a lui adiacenti nel grafo, purché non compaiono già nell'albero come suoi antenati: per esempio i nodi figli della radice sono n7 e n5; le foglie sono il nodo finale n3 o altri nodi che non hanno figli perché tutti i nodi adiacenti (nel grafo) compaiono già tra i loro antenati (nell'albero); i percorsi sono i "rami" che dalla radice dell'albero vanno alle foglie etichettate col nodo finale.

ESERCIZIO 3
Si faccia riferimento al problema ricorrente MOVIMENTO DI PEZZI DEGLI SCHACCHI, pagina 20.

PROBLEMA
In un campo di dimensioni 8×8 un robot si muove come il cavallo nel giuoco degli scacchi; gli sono vietate, però, le mosse nelle direzioni della rosa dei venti comprese nella seguente lista:

[nne,ene,ese,sse],

cioè le mosse del robot in questo problema si riducono a quelle illustrate (col simbolo ♘) nella seguente figura.



Nel campo di gara le caselle della seguente lista sono interdette al robot:

[[3,3],[3,5],[4,2],[4,4],[5,2],[5,3],[5,4],[5,5]]

N.B. Un elemento della lista descrive una casella indicandone le coordinate a partire dallo spigolo in basso a sinistra del campo di gara.
Inoltre, in certe caselle sono presenti dei premi, descritti dalla seguente lista:

[[2,5,8],[4,3,10],[4,6,11],[5,6,12],[3,4,13]].

N.B. Un elemento della lista ha la forma: [<ascissa>,<ordinata>,<premio>].
Partendo dalla casella [8,1], il robot deve raggiungere la casella [1,8], senza passare più di una volta per una stessa casella. Trovare:

1. il numero N1 di percorsi in cui non si raccolgono premi;
2. il numero N2 di percorsi in cui si raccoglie un premio di 11.

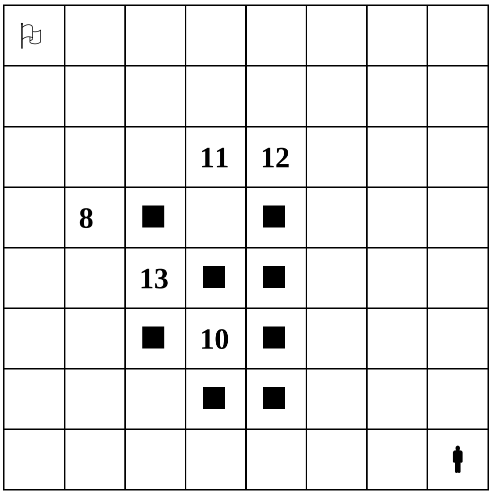
N1	
N2	

SOLUZIONE

N1	3
N2	2

COMMENTI ALLA SOLUZIONE
Esiste una maniera sistematica per trattare problemi di questo tipo: costruire l'*albero delle possibili mosse*. Ogni nodo dell'albero è etichettato con le coordinate di una casella; si inizia con la *radice* che è la casella in cui parte il robot; poi ad ogni nodo si aggiungono tanti *figli* quante sono le caselle raggiungibili dal robot posto nella casella corrispondente a quel nodo. Naturalmente il robot non può tornare in una casella in cui è già stato.
I nodi in cui ci si arresta (cioè le foglie dell'albero) sono la *meta* o una casella da cui il robot non si può muovere.

In casi "semplici", come il presente, si possono costruire direttamente tutti i percorsi possibili.
N.B. Il presente caso è "semplice" perché al robot è permesso di *muoversi solo verso sinistra*, quindi è facile visualizzarne i percorsi: in particolare all'inizio non potrà "salire" troppo, perché altrimenti non può raggiungere la ottava colonna (appunto perché è limitato nei movimenti).
Il campo di gara è mostrato nella figura.



È immediato convincersi che da [8,1] il robot non può andare in [6,2] perché, essendo la casella [5,4] interdetta e dovendo muoversi verso sinistra, può andare solo in [4,3] oppure in [4,1]: da entrambe non può che andare nella seconda colonna (quindi non potrà arrivare in [1,8]).
La prima mossa deve quindi essere in [7,3]; la seconda mossa è obbligata in [6,5]: infatti il robot potrebbe andare anche in [6,1], ma da quest'ultima casella non potrebbe più muoversi.
Da [6,5] il robot può andare solo in [4,6] oppure in [5,7]. È facile vedere dalla prima casella ci sono due percorsi per raggiungere la meta, mentre dalla seconda casella ci sono tre percorsi. Ricapitolando i percorsi possibili e i premi complessivi raccolti sono:

PERCORSI	PREMI RACCOLTI
[[8,1],[7,3],[6,5],[4,6],[3,4],[2,6],[1,8]]	11 + 13
[[8,1],[7,3],[6,5],[4,6],[3,8],[2,6],[1,8]]	11
[[8,1],[7,3],[6,5],[5,7],[4,5],[2,6],[1,8]]	0
[[8,1],[7,3],[6,5],[5,7],[4,5],[3,7],[1,8]]	0
[[8,1],[7,3],[6,5],[5,7],[3,8],[2,6],[1,8]]	0

ESERCIZIO 4
Si faccia riferimento all'Allegato A - OPS 2016, ELEMENTI DI PSEUDOLINGUAGGIO, pagina 23.

PROBLEMA
Si consideri la seguente procedura PROVA1.

```
procedure PROVA1;
variables A, B, K integer;
A ← 1000;
B ← 0;
K ← 0;
while B < A do
        K ← K +2;
        for J from 1 to 4 step 1 do
                K ← K + K;
        endfor;
        B ← K + B;
endwhile;
output K, B;
endprocedure;
```

Determinare il valore di output di K e B.

K	
B	

SOLUZIONE

K	8736
B	9312

COMMENTI ALLA SOLUZIONE
La seguente tabella mostra i valori assunti da K, A, B, prima del ciclo "while" e in relazione a ogni esecuzione dello stesso; le ripetizioni si interrompono quando (il valore di) B supera (il valore di) A.

		valore		
		A	K	B
dopo i primi 3 statement di assegnazione		1000	0	0
prima esecuzione del "while", dopo il primo statement		1000	2	0
	dopo l'esecuzione del "for"	1000	32	0
	dopo l'ultimo statement	1000	32	32
seconda esecuzione del "while", dopo il primo statement		1000	34	32
	dopo l'esecuzione del "for"	1000	544	32
	dopo l'ultimo statement	1000	544	576
terza esecuzione del del "while", dopo il primo statement		1000	546	576
	dopo l'esecuzione del "for"	1000	8736	576
	dopo l'ultimo statement	1000	8736	9312

ESERCIZIO 5
Si faccia riferimento all'Allegato A - OPS 2016, ELEMENTI DI PSEUDOLINGUAGGIO, pagina 23.

PROBLEMA
Si consideri la seguente procedura PROVA2.

```
procedure PROVA2;
variables Q, M, J, K integer;
Q ← 1;
M ← 1;
K ← 1;
for J from 1 to 8 step 1 do
        K ← K + J;
        M ← M × J;
        Q ← Q + K – J;
 endfor;
output M, Q;
endprocedure;
```

Trovare i valori di output per M e Q.

M	
Q	

SOLUZIONE

M	40320
Q	93

COMMENTI ALLA SOLUZIONE
È facile ricavare i valori richiesti schematizzando con una tabella il comportamento del programma.

dopo	valori di			
	J	K	M	Q
i primi 3 statement di assegnazione	indefinito	1	1	1
la prima ripetizione del ciclo for	1	2	1	2
la seconda ripetizione del ciclo for	2	4	2	4
la terza ripetizione del ciclo for	3	7	6	8
la quarta ripetizione del ciclo for	4	11	24	15
la quinta ripetizione del ciclo for	5	16	120	26
la sesta ripetizione del ciclo for	6	22	720	42
la settima ripetizione del ciclo for	7	29	5040	64
la ottava ripetizione del ciclo for	8	37	40320	93

ESERCIZIO 6
Si faccia riferimento all'Allegato A - OPS 2016, ELEMENTI DI PSEUDOLINGUAGGIO, pagina 23.

PROBLEMA
Si consideri la seguente procedura PROVA3.

```
procedure PROVA3;
variables A, Q, M, J, K integer;
Q ← 0;
M ← 0;
for J from 1 to 4 step 1 do
        input A;
        for I from 1 to 8 step 1 do
        if I = J then Q ← Q + A;  endif;
                if I > J then M ← M + A;  endif;
        endfor;
endfor;
output M, Q;
endprocedure;
```

Trovare i valori di output, sapendo che i valori di input per A sono i seguenti: 4,3,2,1 .

M	
Q	

SOLUZIONE

M	60
Q	10

COMMENTI ALLA SOLUZIONE
La procedura accumula in Q ed M i valori letti per A, di volta in volta con una certa molteplicità. Più esattamente, l'evoluzione del valore di Q ed M è mostrato nella seguente tabella.
N.B. Il costrutto "for J …" si può chiamare "for" *esterno* e il costrutto "for I …" si può chiamare "for" *interno*, per ovvie ragioni

azioni	valori di			
	J	A	Q	M
prima esecuzione del "for" esterno	1	4	0	0
dopo l'esecuzione del "for" interno con I che vale 1	1	4	4	0
dopo l'esecuzione del "for" interno con I che vale 2, 3, 4, 5, 6, 7, 8	1	4	4	28
seconda esecuzione del "for" esterno	2	3	4	28
dopo l'esecuzione del "for" interno con I che vale 1	2	3	4	28
dopo l'esecuzione del "for" interno con I che vale 2	2	3	7	28
dopo l'esecuzione del "for" interno con I che vale 3, 4, 5, 6, 7, 8	2	3	7	46
terza esecuzione del "for" esterno	3	2	7	46
dopo l'esecuzione del "for" interno con I che vale 1, 2	3	2	7	46
dopo l'esecuzione del "for" interno con I che vale 3	3	2	9	46
dopo l'esecuzione del "for" interno con I che vale 4, 5, 6, 7, 8	3	2	9	56
terza esecuzione del "for" esterno	4	1	9	56

	dopo l’esecuzione del “for” interno con I che vale 1, 2, 3	4	1	9	56
	dopo l’esecuzione del “for” interno con I che vale 4	4	1	10	56
	dopo l’esecuzione del “for” interno con I che vale 5, 6, 7, 8	4	1	10	60

ESERCIZIO 7

PROBLEM

The sum of the first 100 positive integers is 5050. That is, 1 + 2 + … + 99 + 100 = 5050.
What is the sum of the first 100 positive *odd* integers? Put your answer in the box below.
Remember to put a comma every three digits starting from the right, in every number that has five or more digits.

SOLUTION

10,000

TIPS FOR THE SOLUTION

To sum the first 100 positive odd integers, it is crucial to determine the last one: two ways to do this are the following ones.

– The integers 1, 2, 3, …, 199, 200 are (the first) *two hundred* integers: half of them are even and half are odd: this means that the 100-th odd integer is 199.
– Odd integers are obtained from the formula $i \times 2 - 1$, in which i assumes the values 1, 2, … and so forth; for $i = 100$ the formula gives 199.

To compute the sum, the famous Gauss's trick can be used: it is easier to compute twice the sum. Indeed,

$$\begin{array}{l} \ \ 1 + \ 3 \ + \ldots + 197 + 199 + \\ 199 + 197 + \ldots + \ 3 \ + \ 1 \qquad = \\ \hline 200 + 200 + \ldots + 200 + 200 \qquad = 200 \times 100 = 20{,}000 \end{array}$$

Hence the solution is 20,000/2 = 10,000.
Alternatively, one can write and run a simple program.

ESERCIZIO 8

PROBLEM
For every John's birthday, his mother prepares a cake with candles showing his age on top. Since his fourth birthday, John has always blown out all the candles. Before that age, he averaged a 50% total blowout rate. So far, John has blown out exactly 900 candles. How old is he? Put your answer in the box below.

SOLUTION

42

TIPS FOR THE SOLUTION
On his first three birthdays John was supposed to blow out 1 + 2 + 3 = 6 candles; as his success rate was 50% he missed three candles. Hence the total number of candles on the birthday cakes was 903; if we call *n* the age of John, then 1 + 2 + 3 + … + *n* = 903.
Remembering the Gauss's trick, we have

$$\frac{n \times (n+1)}{2} = 903$$

that is

$$n^2 + n - 1806 = 0$$

The solution is

$$n = \frac{-1 \pm \sqrt{1 + 4 \times 1806}}{2}$$

$$n = \frac{-1 + 85}{2} = 42$$

(Of course the negative solution was discarded.)