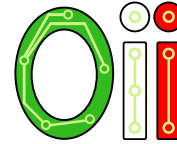




International Olympiad in Informatics



Olimpiadi Italiane di Informatica

## OLIMPIADI DI INFORMATICA 2018-2019 SELEZIONE SCOLASTICA – 15 novembre 2018

### ISTRUZIONI PER LO SVOLGIMENTO DELLA PROVA

- 1) La prova consiste di **5 esercizi a carattere logico matematico, 7 esercizi di programmazione e 8 esercizi a carattere algoritmico**. Il tempo a disposizione per la prova è piuttosto limitato per cui si suggerisce al candidato di non fermarsi a lungo su un esercizio se non riesce a trovarne la soluzione ed eventualmente riprenderlo in esame quando avrà terminato di eseguire tutti gli esercizi successivi.
- 2) Gli esercizi sono di due tipi: a risposta chiusa, con domande seguite da quattro possibili alternative (indicate con le lettere a, b, c, d) di cui una sola è corretta; a risposta aperta, quando è richiesto che la soluzione venga scritta direttamente dal candidato.
- 3) A ogni esercizio è associato un punteggio correlato al livello di difficoltà. Il punteggio è indicato all'inizio dell'esercizio ed è ripetuto nella tabella delle risposte. La valutazione viene effettuata come segue:
  - a ogni risposta esatta viene assegnato il punteggio corrispondente;
  - a ogni risposta sbagliata viene assegnato un punto negativo nel caso di esercizi a risposta chiusa, cioè con scelta tra più alternative;
  - a ogni risposta sbagliata vengono assegnati zero punti nel caso di esercizi a risposta aperta, cioè con soluzione scritta direttamente dal candidato;
  - a ogni esercizio lasciato senza risposta vengono assegnati zero punti.
- 4) La risposta va riportata nell'apposito spazio della tabella delle risposte segnando il quadratino corrispondente a quella ritenuta esatta, oppure scrivendola per esteso, nel caso la domanda sia a risposta aperta. Su tale tabella non sono ammesse cancellature o correzioni, pena l'invalidazione della prova.
- 5) Non è consentito l'uso di alcun dispositivo elettronico (palmare, telefono, etc.). Non è permesso consultare libri, appunti, manuali, pena l'esclusione dalla selezione. È consentito solo utilizzare fogli bianchi per appunti e calcoli.
- 6) Il tempo assegnato per svolgere la prova è di **90 minuti**.
- 7) Il candidato è tenuto a indicare chiaramente sulla scheda nome, cognome, data di nascita, classe, linguaggio di programmazione scelto e, se lo possiede, indirizzo e-mail.



**AICA**  
Associazione Italiana per l'Informatica  
ed il Calcolo Automatico



Ministero dell'Istruzione  
dell'Università e della Ricerca

## SCHEDA STUDENTI

Inserire i propri dati anagrafici nei campi seguenti.

I campi preceduti da (\*) sono obbligatori.

**\* Nome:**

**\* Cognome:**

**\* Data di nascita:**

**\* Classe:**

**Linguaggio:**

**Codice Fiscale:**

**\* E-mail:**

## ESERCIZI A CARATTERE LOGICO-MATEMATICO

---

**Esercizio N° 1 – La risposta esatta vale 1 punto.**

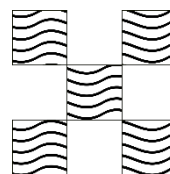
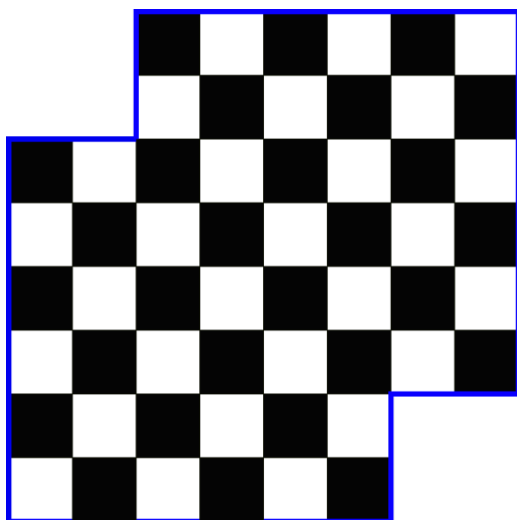
Hai prestato un libro al tuo amico Giulio, ma, quando te lo ha riportato, mancavano le pagine 7, 8, 100, 101, 222 e 223. Qual è il minimo numero **NUMFOGLI** di fogli che ha strappato?

**NUMFOGLI =**

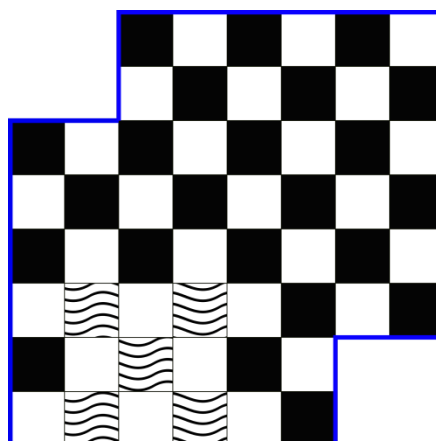
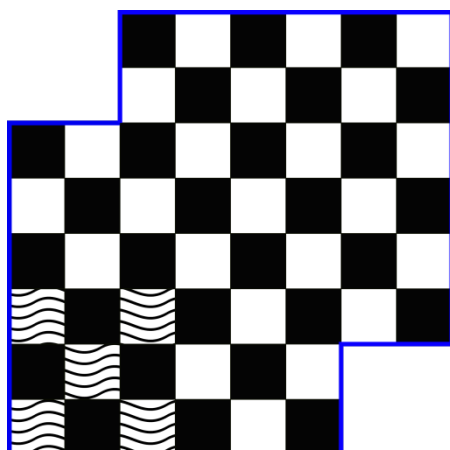
---

**Esercizio N° 2 – La risposta esatta vale 1 punto.**

Data la seguente "scacchiera mutilata", dire qual è il numero **NUMPOS** di posizioni diverse in cui è possibile inserire la tesserina a lato.



Esempi corretti di inserimento della tesserina:



**NUMPOS =**

---

---

**Esercizio N° 3 – La risposta esatta vale 2 punti.**

Giulia, da quando ha imparato le percentuali a scuola, parla soltanto esprimendo ogni valore quantitativo mediante percentuale. Sappiamo che:

- la scuola ha comprato una cassa da 3kg (lordi) di mele;
- la tara era il 20%;
- ogni scolaro poteva mangiare lo 0,8% del peso netto delle mele;
- Giulia, amica della bidella, ha ricevuto il 200% della razione degli altri.

La mamma di Giulia, che non era molto brava a scuola, vorrebbe capire quanti chilogrammi di mele ha mangiato sua figlia in mensa:

- (a) 48/625 kg
- (b) 24/625 kg
- (c) 3/625 kg
- (d) 6/625 kg
- (e) Non rispondo

---

---

**Esercizio N° 4 – La risposta esatta vale 2 punti.**

Le due sorelle Anna e Zoe quando vogliono parlare dei loro segreti fanno questo gioco: hanno tre scalini sul portico di casa e, a seconda dello scalino sul quale si trovano, devono dire la verità o una bugia.

Anna sale sul primo gradino ed afferma: "La mia bicicletta non è rotta".

Zoe sale sul secondo gradino e dice: "La tua bicicletta l'ho rotta io".

Anna sale di un gradino e dice: "Tu hai rotto la mia bici".

Zoe sale ancora di uno scalino e replica: "La tua bicicletta è rotta".

Sapendo che c'è esattamente un gradino dove viene detta la verità scegliere quale tra le seguenti affermazioni è sbagliata:

- (a) Due tra le seguenti affermazioni sono corrette
- (b) "Gradino 1 -> verità" è una scelta che non genera contraddizioni
- (c) "Gradino 3 -> verità" è una scelta che non genera contraddizioni
- (d) "Gradino 2 -> verità" è una scelta che non genera contraddizioni
- (e) Non rispondo

---

---

**Esercizio N° 5 – La risposta esatta vale 3 punti.**

La differenza simmetrica di due insiemi A e B è l'insieme  $A \Delta B = (A \cup B) - (A \cap B)$ , dove  $\cup$  è il simbolo dell'unione insiemistica, mentre  $\cap$  è il simbolo dell'intersezione.

Se A e B sono i due insiemi seguenti:

$A = \{2 \leq x \leq 30 : x \equiv 2 \pmod{7}\}$ , dove  $x \equiv 2 \pmod{7}$  significa che x dà resto 2 se diviso per 7

$B = \{2 \leq x \leq 20 : x \text{ non è primo}\}$

Quali sono gli elementi contenuti nell'insieme **INS** =  $A \Delta B$  ?

**INS =**

(Nota: inserire gli elementi (senza usare "leading zeros") separandoli con una virgola, senza lasciare spazi tra di essi. Es. 9,11,56,...)

## ESERCIZI DI PROGRAMMAZIONE

Gli esercizi di programmazione saranno specificati in **pseudocodice** al fine di valutare le capacità di programmazione prescindendo dalla conoscenza di uno specifico linguaggio. Di seguito vengono elencate alcune delle possibili forme sintattiche. Qualora un esercizio risultasse ambiguo, la consegna stessa conterrà delle note atte a chiarire eventuali dubbi.

<b>Pseudocodice</b>	<b>Significato</b>
a ← 3; a ← a + 1;	assegnamento del valore 3 ad a, incremento del valore di a
leggi(a); scrivi(a);	istruzioni per leggere dati da tastiera e scrivere dati sullo schermo
scrivi("ciao"); scrivi("ciao", var);	scrive la stringa ciao a video scrive ciao e poi il contenuto della variabile var
istruzione;	il punto e virgola, per convenzione, indica la fine di un'istruzione
array[7] ← 0; mat[5][28] ← 1;	esempi di inizializzazione di un elemento di un array e di una cella di una matrice
+ - * / div mod	questi simboli indicano rispettivamente le operazioni aritmetiche di somma, differenza, prodotto, divisione, divisione intera (quoziente) e resto della divisione
= < > ≤ ≥ ≠	sono i simboli che indicano i sei tipi di confronto: uguale, minore, maggiore, minore o uguale, maggiore o uguale, diverso
(a ≤ 3) <b>oppure</b> (a ≥ 5) (a ≠ 0) <b>e</b> (b ≠ 0) <b>non</b> (a)	esempi di operazioni logiche: <b>non</b> per il NOT logico, <b>e</b> per l'AND, <b>oppure</b> per l'OR
<b>se</b> condizione <b>allora</b> ... <b>fine se</b>	è un costrutto condizionale
<b>finché</b> condizione <b>esegui</b> ... <b>fine finché</b>	è un ciclo a condizione iniziale (pre-posta)
<b>ripeti</b> ... <b>finché</b> condizione	è un ciclo a condizione finale (post-posta)
<b>funzione</b> funzione(p1: tipo, p2: tipo): tipo ... <b>restituisce</b> valore_restituito; <b>fine funzione</b>	sintassi di una funzione che prende due parametri p1 e p2, e restituisce un certo valore
<b>procedura</b> procedura(p1: tipo, p2: tipo) ... <b>fine procedura</b>	sintassi di una procedura (funzione che non restituisce un valore)

I "blocchi di codice" sono indicati dall'indentazione, ovvero dalla quantità di spazi presenti ad inizio riga.

---

**Esercizio N° 6 – La risposta esatta vale 1 punto.**

Dato il seguente codice:

```
1: conta ← 0;
2: alfa ← 0;
3: beta ← 0;
4: finché conta < 29 esegui
5:     se (conta mod 3) = 1 allora
6:         alfa ← alfa + 2;
7:     altrimenti
8:         beta ← beta - 1;
9:     fine se
10:    conta ← conta + 2;
11: fine finché
```

calcolare il numero **NUMRIP** di volte per cui viene eseguito il ciclo **finché...esegui**. In altre parole, calcolare il numero di volte che viene eseguito il blocco di codice tra le righe 5 e 10.

**NUMRIP =**

*Nota:* si ricorda che con  $(a \bmod b)$  va inteso il resto ottenuto dividendo a per b.

---

**Esercizio N° 7 – La risposta esatta vale 2 punti.**

Ti viene data la seguente funzione, dove:

- il parametro a è un array di numeri interi indicizzato a partire da 1,
- il parametro N è un numero intero che rappresenta la dimensione dell'array a

```
1: funzione fun(a: array di interi, N: intero): intero
2:     palo ← -1;
3:     dado ← -1;
4:     i ← 1;
5:     finché i ≤ N esegui
6:         se (a[i] mod 2) = 0 allora
7:             se a[i] > palo allora
8:                 palo ← a[i];
9:             altrimenti
10:                se a[i] > dado allora
11:                    dado ← a[i];
12:            fine se
13:        fine se
14:    fine se
15:    i ← i + 1;
16: fine finché
17: restituisce (palo + dado);
18: fine funzione
```

indicare quale tra le seguenti affermazioni è FALSA:

- (a) La funzione `fun` restituisce 23 se riceve in ingresso: `a={1, 2, 6, 10, 22}`, ed: `N=5`  
 (b) La funzione `fun` restituisce la massima somma di due elementi dell'array  
 (c) La funzione `fun` restituisce numeri sia pari sia dispari  
 (d) La funzione `fun` non può restituire valori inferiori a -2  
 (e) Non rispondo

**Esercizio N° 8 – La risposta esatta vale 2 punti.**

Si vuole scrivere una procedura che visualizzi a video una griglia 5×5 di numeri avente il contenuto seguente:

0	0	0	0	0
0	1	1	1	0
0	0	1	0	0
0	0	1	0	0
0	0	0	0	0

Il codice qui sotto dà per scontato:

- che sia dichiarata una matrice `mat` di dimensione 5×5 composta da numeri interi, con gli indici che partono da 1 sia per le righe che per le colonne,
- che per mandare a capo la stampa su schermo si possa usare l'istruzione `scrivi("↓")`

```

1: mat[5][5] ← 0;
2: i ← 1;
3: finché i ≤ 5 esegui
4:     i ← i + 1;
5:     j ← 1;
6:     finché j ≤ 5 esegui
7:         /// QUI MANCA DEL CODICE ///
8:         /// QUI MANCA DEL CODICE ///
9:         /// QUI MANCA DEL CODICE ///
10:        /// QUI MANCA DEL CODICE ///
11:        /// QUI MANCA DEL CODICE ///
12:        /// QUI MANCA DEL CODICE ///
13:        /// QUI MANCA DEL CODICE ///
14:        /// QUI MANCA DEL CODICE ///
15:        /// QUI MANCA DEL CODICE ///
16:        /// QUI MANCA DEL CODICE ///
17:        /// QUI MANCA DEL CODICE ///
18:        /// QUI MANCA DEL CODICE ///
19:        /// QUI MANCA DEL CODICE ///
20:        j ← j + 1;
21:     fine finché
22: fine finché
23: i ← 1;
24: finché i ≤ 5 esegui
25:     j ← 1;
26:     finché j ≤ 5 esegui
  
```

```

27:     scrivi(mat[i][j]);
28:     j ← j + 1;
29:     fine finché
30:     scrivi("↓");
31:     i ← i + 1;
32: fine finché

```

Nel codice sopra, come si può notare, manca un pezzo. Quale tra i seguenti pezzi è quello da inserire?

(a)

```

7:  se i-1 ≠ 0 allora
8:    se i-1 = 1 allora
9:      se (j ≠ 0) e (j ≠ 4) allora
10:        mat[i-1][j] ← 1;
11:      fine se
12:    fine se
13:    se (i-1 = 2) e (j = 2) allora
14:      mat[i-1][j] ← 1;
15:    fine se
16:    se (i-1 = 2) e (j = 3) allora
17:      mat[i-1][j] ← 1;
18:    fine se
19:  fine se

```

(b)

```

7:  se i-1 ≠ 0 allora
8:    se i = 1 allora
9:      se (j ≠ 0) e (j ≠ 4) allora
10:        mat[i][j] ← 1;
11:      fine se
12:    fine se
13:    se (i = 2) e (j = 2) allora
14:      mat[i][j] ← 1;
15:    fine se
16:    se (i = 3) e (j = 2) allora
17:      mat[i][j] ← 1;
18:    fine se
19:  fine se

```

(c)

```

7:  se i-1 ≠ 0 allora
8:    se i-1 = 1 allora
9:      se (j ≠ 1) e (j ≠ 5) allora
10:        mat[i-1][j] ← 1;
11:      fine se
12:    fine se
13:    se (i-1 = 2) e (j = 2) allora
14:      mat[i-1][j] ← 1;
15:    fine se
16:    se (i-1 = 3) e (j = 2) allora
17:      mat[i-1][j] ← 1;
18:    fine se
19:  fine se

```

(d)

```

7:  se i-1 ≠ 0 allora
8:    se i-1 = 1 allora
9:      se (j ≠ 0) e (j ≠ 4) allora
10:        mat[i-1][j] ← 1;
11:      fine se
12:    fine se
13:    se (i-1 = 2) e (j = 2) allora
14:      mat[i-1][j] ← 1;
15:    fine se
16:    se (i-1 = 3) e (j = 2) allora
17:      mat[i-1][j] ← 1;
18:    fine se
19:  fine se

```

(e) *Non rispondo*



---

**Esercizio N° 9 – La risposta esatta vale 2 punti.**

Date le seguenti due funzioni:

```
1: funzione mystery(a: intero, b: intero): intero
2:     se (2 * a) > b allora
3:         restituisce b;
4:     altrimenti
5:         restituisce a;
6:     fine se
7: fine funzione
8:
9: funzione secret(a: intero, b: intero): intero
10:    se (a + b) > mystery(a, b) allora
11:        restituisce a;
12:    altrimenti
13:        restituisce mystery(a, b);
14:    fine se
15: fine funzione
```

indicare quale valore **RES** viene restituito dalla chiamata secret(24, 3)

**RES =**

---

**Esercizio N° 10 – La risposta esatta vale 2 punti.**

Dato il seguente pseudocodice:

```
1: funzione fun(fiore: intero, farfalla: intero): intero
2:     se fiore = farfalla allora
3:         restituisce farfalla;
4:     altrimenti
5:         se farfalla > fiore allora
6:             restituisce fun(farfalla - fiore, fiore);
7:         altrimenti
8:             restituisce fun(farfalla, farfalla - fiore);
9:         fine se
10:    fine se
11: fine funzione
```

Cosa si può dire della funzione fun?

- (a) Non termina per alcun valore della coppia: (fiore, farfalla)
- (b) Non termina soltanto quando: fiore = farfalla
- (c) Termina sicuramente quando: fiore = farfalla
- (d) Termina per ogni valore in cui: farfalla > fiore
- (e) Non rispondo

---

**Esercizio N° 11 – La risposta esatta vale 3 punti.**

Si considerino le seguenti due funzioni, che prendono in ingresso un numero intero maggiore o uguale a zero:

```
1: funzione effe(a: intero): intero
2:     se a < 2 allora
3:         restituisce 2 * a;
4:     altrimenti
5:         restituisce gi(a - 1) + gi(a - 2);
6:     fine se
7: fine funzione
8:
9: funzione gi(a: intero): intero
10:    se a < 2 allora
11:        restituisce effe(a);
12:    altrimenti
13:        restituisce effe(a - 1) + effe(a - 2);
14:    fine se
15: fine funzione
```

Indicare quale valore **RES** viene restituito dalla chiamata effe(10)

**RES =**

---

**Esercizio N° 12 – La risposta esatta vale 3 punti.**

Data la seguente funzione:

```
1: funzione calc(n: intero): intero
2:     se n = 1 allora
3:         restituisce 2;
4:     altrimenti
5:         restituisce 2 * n - 1 + calc(n - 1);
6:     fine se
7: fine funzione
```

indicare quale tra le seguenti espressioni è il valore che viene restituito se  $n \geq 1$

- (a)  $(n+1)^2$
- (b)  $n^2$
- (c)  $(n-1)^2$
- (d)  $n^2+1$
- (e) *Non rispondo*

## ESERCIZI A CARATTERE ALGORITMICO

---

### Esercizio N° 13 – La risposta esatta vale 1 punto.

Nel palacongressi di Audiola ci sono quattro casse acustiche che indichiamo con i numeri 1, 2, 3 e 4. A ogni spettatore  $s_n$  è associato un vettore di 4 componenti (numeri naturali) che ne esprime la distanza dalle 4 casse. Lo spettatore sente cosa riproduce la cassa  $i$  se e solo se l' $i$ -esima componente del suo vettore è minore di  $i^2$ .

Le casse regolano il proprio volume in base ad una semplice regola: ciò che viene riprodotto da una cassa non può essere udito da più di 3 spettatori contemporaneamente. Ciascuno spettatore può invece sentire anche da più di una cassa. Il totale di spettatori è 15, dei quali precisamente 4 non sentono alcun suono. Dei 3 spettatori  $s_{13}$ ,  $s_{14}$ ,  $s_{15}$  il vettore è ignoto e deve essere ricostruirlo, mentre gli altri sono:

$s_1 \rightarrow (1, 4, 3, 4)$   
 $s_2 \rightarrow (14, 25, 17, 19)$   
 $s_3 \rightarrow (18, 21, 38, 17)$   
 $s_4 \rightarrow (1, 4, 9, 16)$   
 $s_5 \rightarrow (0, 5, 10, 17)$   
 $s_6 \rightarrow (0, 15, 36, 23)$   
 $s_7 \rightarrow (1, 6, 10, 15)$   
 $s_8 \rightarrow (2, 2, 10, 16)$   
 $s_9 \rightarrow (3, 3, 11, 17)$   
 $s_{10} \rightarrow (0, 15, 16, 17)$   
 $s_{11} \rightarrow (1, 5, 8, 19)$   
 $s_{12} \rightarrow (1, 5, 7, 44)$

Quale delle seguenti ipotesi di vettori per i tre spettatori restanti NON è possibile?

- (a)  $s_{13} \rightarrow (1, 5, 10, 17)$   
 $s_{14} \rightarrow (1, 3, 10, 23)$   
 $s_{15} \rightarrow (1, 7, 15, 15)$
- (b)  $s_{13} \rightarrow (2, 6, 11, 18)$   
 $s_{14} \rightarrow (1, 3, 10, 24)$   
 $s_{15} \rightarrow (1, 7, 15, 14)$
- (c)  $s_{13} \rightarrow (0, 4, 9, 16)$   
 $s_{14} \rightarrow (1, 3, 10, 25)$   
 $s_{15} \rightarrow (1, 7, 15, 13)$
- (d)  $s_{13} \rightarrow (3, 7, 12, 19)$   
 $s_{14} \rightarrow (1, 3, 10, 26)$   
 $s_{15} \rightarrow (1, 7, 15, 12)$
- (e) *Non rispondo*

---

### Esercizio N° 14 – La risposta esatta vale 1 punto.

Elon Musk si sta organizzando per andare su Marte con la sua famiglia. Tuttavia, per minimizzare i rischi, la navicella porta al massimo una persona (oltre al pilota). Le richieste da parte della famiglia di Elon sono:

- Maye (la mamma di Elon) guarda il panorama e non vuole metterci meno di 20 giorni;
- Errol (il papà di Elon) non ha nessuna paura della velocità ed è anche in grado di pilotare la navicella. Impiega 1 giorno a raggiungere Marte dalla Terra e viceversa;

- Kimbal (fratello di Elon) sa guidare la navicella, ma non a una velocità troppo elevata, quindi, se guida, ci mette almeno 40 giorni;
- Tosca (sorella di Elon) non vuole impiegare meno di 25 giorni per il viaggio.

Sapendo che Elon sa guidare la navicella e si rifiuta di raggiungere velocità stratosferiche, in particolare non vuole che il viaggio duri meno di 15 giorni, qual è il tempo minimo  $T_{min}$  perché tutta la famiglia di Elon arrivi su Marte?

$T_{min} =$

### Esercizio N° 15 – La risposta esatta vale 2 punti.

Il caveau di una banca è protetto da una password che può essere digitata su un tastierino numerico di 10 cifre (da 0 a 9). La cassaforte è dotata di 4 luci di colore diverso che hanno il seguente comportamento:

- GIALLO: finora tutto OK, proseguire pure;
- BLU: mancano tante cifre quante quelle già inserite;
- ROSSO: è stato commesso un errore: ricominciare dall'inizio;
- VERDE: combinazione corretta, apertura porta in corso.

Eve vuole svaligiare la banca e ha accumulato le seguenti informazioni sui tentativi di apertura del caveau da parte dei dipendenti:

1. CIFRA (ignota) - luce gialla; CIFRA (ignota) - luce gialla; CIFRA (ignota) - luce blu; CIFRA (7) - luce gialla; CIFRA (8) - luce gialla; CIFRA (ignota) - non vede la luce
2. CIFRA (ignota) - luce gialla; CIFRA (ignota) - luce gialla; CIFRA (7) - luce rossa
3. CIFRA (ignota) - luce gialla; CIFRA (7) - luce rossa
4. CIFRA (2) luce gialla - CIFRA (ignota) - luce gialla; CIFRA (3) - luce blu; CIFRA (4) - luce rossa

Quale tra le seguenti affermazioni di Eve è l'unica corretta?

- (a) La password potrebbe contenere più di 6 caratteri
- (b) La password contiene non contiene cifre 7
- (c) La password contiene come prima cifra 2
- (d) La password non contiene cifre pari
- (e) Non rispondo

### Esercizio N° 16 – La risposta esatta vale 2 punti.

In una scuola nel giorno della memoria vengono proiettati 5 diversi film  $f_n$  legati alla seconda guerra mondiale e gli studenti  $s_m$  delle varie classi votano per scegliere quale film guardare.

Una classe va a vedere un film se:

- non più di 3 studenti sono contrari a quel film e
- fra i film che rimangono, il numero di studenti interessati al film è massimo e
- fra i film che rimangono, il numero di studenti contrari è minimo.

Le preferenze espresse dagli studenti della classe IIA sono le seguenti:

Associazioni positive (studenti favorevoli):

$(s_1, f_1)$   $(s_2, f_2)$   $(s_3, f_1)$   $(s_4, f_2)$   $(s_5, f_1)$   $(s_6, f_3)$   $(s_7, f_1)$

Associazioni negative (studenti contrari):









$(s_7, f_1)$   $(s_8, f_1)$   $(s_3, f_1)$   $(s_4, f_2)$   $(s_5, f_4)$   $(s_6, f_1)$   $(s_3, f_2)$   $(s_6, f_2)$   $(s_{12}, f_3)$   
 $(s_{11}, f_2)$   $(s_9, f_5)$

Quale è il codice **IDFILM** del film che andrà a vedere la classe IIA?


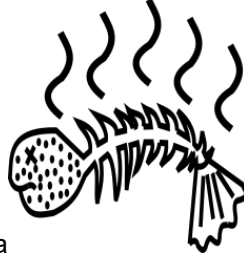
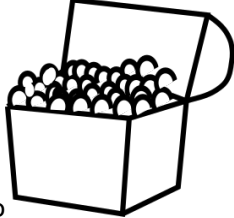
**IDFILM =**

**Esercizio N° 17 – La risposta esatta vale 2 punti.**

Nel mondo del Wumpus un guerriero vuole raggiungere il tesoro. Ma c'è un nemico da evitare: il Wumpus. Il Wumpus è invisibile e il guerriero può solo percepire la sua presenza dalla puzza che emana.

G F E D C B A							
							
							
							
							
	1	2	3	4	5	6	7

Legenda:

 Il guerriero  
 La puzza  
 Il tesoro

Se la puzza si trova in posizione  $(x, y)$  questo significa che il Wumpus si può trovare in una delle quattro posizioni  $(x+1, y)$   $(x-1, y)$   $(x, y+1)$   $(x, y-1)$ .

Il guerriero non vuole rischiare di incontrare il Wumpus (quindi evita le caselle adiacenti alla puzza), si può muovere in alto, in basso, a destra e a sinistra, ma **non** in diagonale.

Sapendo che un percorso viene indicato come una successione di coppie lettera-numero (per es.: A1 B1 C1) scrivere il percorso più breve **PERCBRE** che deve fare il guerriero per arrivare al tesoro.

**PERCBRE =**

(Nota: inserire la successione di coppie lettera-numero, separandola con un trattino, iniziando da A1. Es. A1-B1-C1,...)

**Esercizio N° 18 – La risposta esatta vale 2 punti.**

Alice e Bob si stanno scambiando messaggi cifrati e Hack ha intercettato un messaggio:

“GONSIUZMS ITTO VZGO”

Hack ha scoperto che Alice e Bob usano il cifrario di Cesare (traslano l'alfabeto italiano di una costante  $k$ , per es. se  $k=1$  A diventa B, B diventa C, e così via fino a Z che diventa A).

Alfabeto: A, B, C, D, E, F, G, H, I, L, M, N, O, P, Q, R, S, T, U, V, Z. Trova la costante  $k$ , per rispondere alla seguente domanda:

quale è il messaggio **MESS** scambiato da Alice e Bob?

**MESS =**

(Nota: inserire il messaggio senza le virgolette)

---

---

**Esercizio N° 19 – La risposta esatta vale 3 punti.**

La grafica della tartaruga prevede che si possano impartire degli ordini di movimento a una tartaruga, che li eseguirà lasciando sul terreno una traccia dei suoi movimenti, come se avesse una penna attaccata sulla pancia. Gli ordini possono essere impartiti tramite un semplice linguaggio, stando attenti che:

- le istruzioni *destra* e *sinistra* sono relative all'orientamento attuale della tartaruga, e il numero che segue è un angolo di rotazione (rispettivamente orario e antiorario) espresso in gradi;
- le istruzioni *pennasu* e *pennagiu* sollevano e abbassano rispettivamente la penna sotto la pancia della tartaruga: quando la penna è sollevata ovviamente non lascia tracce sul terreno;
- l'istruzione *ripeti* fa ripetere il blocco che segue, delimitato da parentesi graffe, per un numero di volte indicato a fianco dell'istruzione.

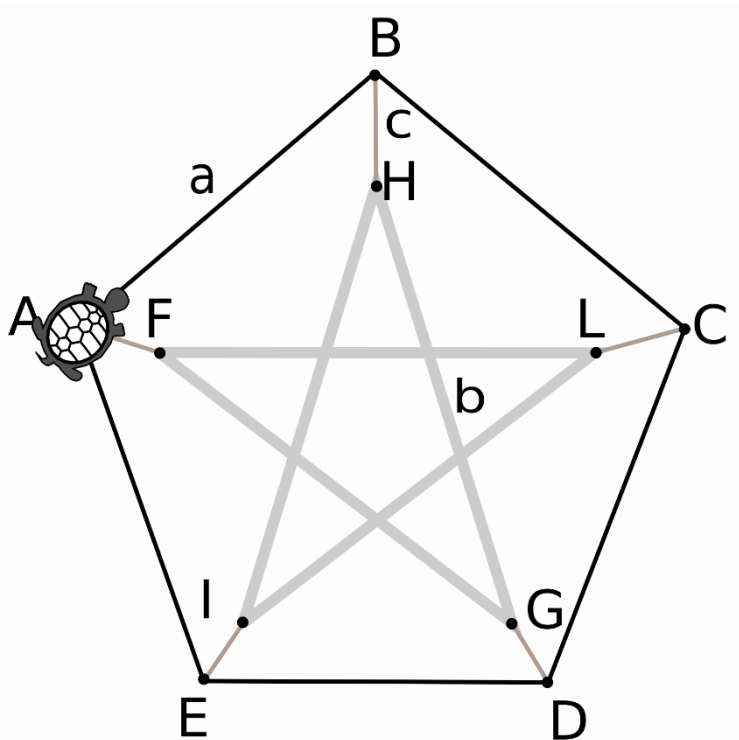
Inizialmente la tartaruga si trova nel vertice A, guarda in alto a destra ed è nella condizione *pennagiu*.

La misura dei lati è  $AB = a$ ,  $BH = c$ ,  $GH = b$ .

Il programmatore della tartaruga è stato però interrotto nel suo lavoro prima di poter scrivere le ultime due istruzioni **ISTR18** e **ISTR19** necessarie perché la tartaruga completi il disegno (il famoso grafo di Petersen).

Scrivere le due istruzioni mancanti.

```
[1] pennagiu
[2] ripeti 5 {
[3]   avanti a
[4]   destra 72
[5] }
[6] destra 54
[7] avanti c
[8] ripeti 4 {
[9]   sinistra 18
[10]  avanti b
[11]  sinistra 18
[12]  avanti c
[13]  pennasu
[14]  sinistra 180
[15]  avanti c
[16]  pennagiu
[17] }
[18] ISTR18
[19] ISTR19
```



**ISTR18 =**

**ISTR19 =**

---

---

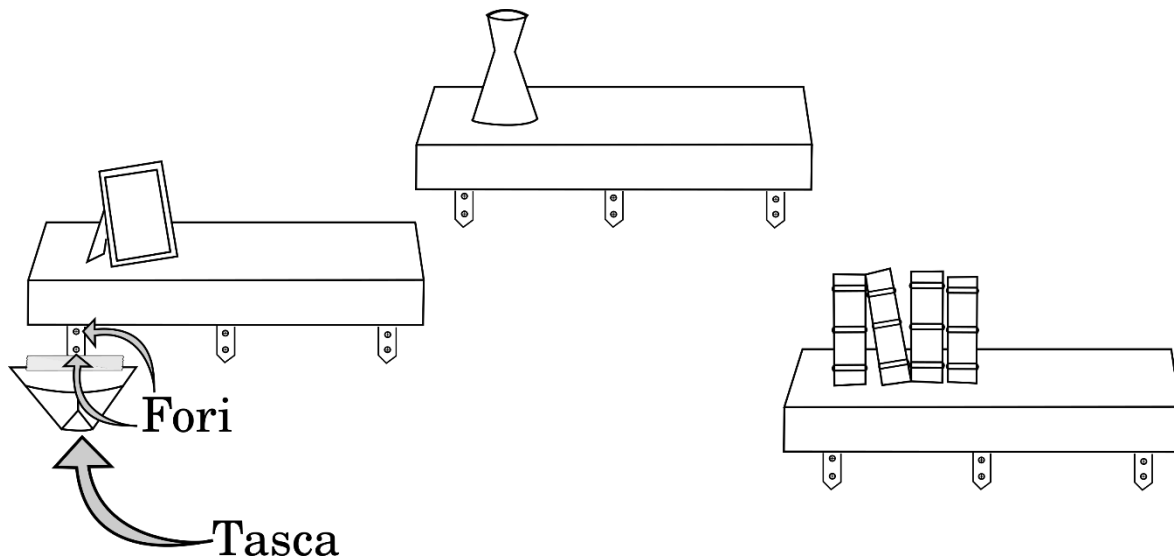
**Esercizio N° 20 – La risposta esatta vale 3 punti.**

Quando Ermelinda, la moglie di Armando, ha deciso di cambiare l'arredamento di casa ha costretto il marito a effettuare il montaggio delle mensole.

Per fissare le mensole al muro, Armando segue questa procedura: per ogni foro per prima cosa ne fa uno molto piccolo (4mm) per poi ingrandirlo (a 8mm). Ogni volta che cambia la punta del trapano (per effettuare fori di diversa dimensione) Armando perde un sacco di tempo, quindi preferisce fare prima tutti i fori piccoli per poi ingrandirli.

Armando è anche molto attento a non sporcare casa con la polvere di mattone e utilizza la cosiddetta "tasca", ossia un cono di carta di giornale attaccato al muro (sotto il foro) grazie a un pezzo di scotch. A ogni spostamento della tasca, però, lo scotch diventa meno adesivo, quindi l'obiettivo di Armando è quello di effettuare tutti i fori spostando il minor numero possibile di volte la tasca.

La figura indica la disposizione delle mensole da appendere (per ogni mensola sono necessari 6 fori).



Sapendo che per ogni coppia di fori (uno sopra e uno sotto ogni staffa di montaggio, come mostrato nella figura) non è necessario spostare la tasca e che inizialmente la tasca si trova sotto i fori di sinistra, qual è il minimo numero **SPOSTmin** di spostamenti della tasca necessari?

E se le coppie di fori fossero  $N$  per mensola ed il numero di mensole fosse  $M$ , quale sarebbe il minimo numero di spostamenti **SPOSTminNM**, scrivendo l'espressione con  $N$  e  $M$  maiuscole e senza spazi?

**SPOSTmin =**

**SPOSTminNM =**