

OLIMPIADI DI INFORMATICA 2019-2020 SELEZIONE SCOLASTICA – 20 novembre 2019

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, genere e indirizzo e-mail.

SCHEMA STUDENTI

Nome: _____ Cognome: _____ Data di nascita: _____

Genere (M/F):

Classe: _____ E-mail: _____

Esercizi a carattere logico-matematico:

Esercizio	Punti	Risposta
1	1	VALTOT =
2	1	NUMCALZINI =
3	2	<input type="checkbox"/> a <input type="checkbox"/> b <input type="checkbox"/> c <input type="checkbox"/> d
4	2	<input type="checkbox"/> a <input type="checkbox"/> b <input type="checkbox"/> c <input type="checkbox"/> d
5	3	<input type="checkbox"/> a <input type="checkbox"/> b <input type="checkbox"/> c <input type="checkbox"/> d

Esercizi di programmazione:

Esercizio	Punti	Risposta
6	1	<input type="checkbox"/> a <input type="checkbox"/> b <input type="checkbox"/> c <input type="checkbox"/> d
7	2	<input type="checkbox"/> a <input type="checkbox"/> b <input type="checkbox"/> c <input type="checkbox"/> d
8	2	<input type="checkbox"/> a <input type="checkbox"/> b <input type="checkbox"/> c <input type="checkbox"/> d
9	2	<input type="checkbox"/> a <input type="checkbox"/> b <input type="checkbox"/> c <input type="checkbox"/> d
10	2	<input type="checkbox"/> a <input type="checkbox"/> b <input type="checkbox"/> c <input type="checkbox"/> d
11	3	NUMREC1 = NUMREC2 =
12	3	A = E = G = H = N = P = R = S = U =

Esercizi a carattere algoritmico:

Esercizio	Punti	Risposta
13	1	DIST =
14	1	c =
15	2	<input type="checkbox"/> a <input type="checkbox"/> b <input type="checkbox"/> c <input type="checkbox"/> d
16	2	NUMMOSSE =
17	2	w =
18	2	<input type="checkbox"/> a <input type="checkbox"/> b <input type="checkbox"/> c <input type="checkbox"/> d
19	3	TOTCOSTO =
20	3	RA = CA = RB = CB =

ESERCIZI A CARATTERE LOGICO-MATEMATICO

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

Valerio e Martina hanno scoperto di aver ereditato una piccola somma da un lontano parente venuto a mancare da poco. Il testamento contiene le indicazioni sull'importo che spetta a ciascuno dei due ragazzi, ovvero quanto segue:

“Vorrei che il doppio di quanto spetti a Martina sia pari al triplo di quanto spetti a Valerio; vorrei inoltre che il doppio di quanto spetti a Valerio, sommato con il triplo di quanto spetti a Martina, sia pari a 13.000 euro.”

Qual è il valore complessivo **VALTOT** dell'eredità di Martina e Valerio?

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

Dato un cassetto con 50 calzini bianchi e 50 calzini neri qual è il numero minimo **NUMCALZINI** di calzini da estrarre per essere sicuri di averne almeno due dello stesso colore?

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

Dato il seguente insieme $A = \{1, 2, 3, 4, 7, 32, 89, 145, 106, 33, 36, 39\}$, qual è il numero di possibili coppie non ordinate di insiemi A_1 e A_2 tali che $|A_1| = |A_2|$ (dove con $|X|$ si intende il numero di elementi contenuti nell'insieme X), $A_1 \cup A_2 = A$, $A_1 \cap A_2 = \emptyset$ e $\text{somma}(A_1) = \text{somma}(A_2)$ (dove $\text{somma}(X)$ è la somma di tutti gli elementi nell'insieme X)? Indicare quella corretta fra le seguenti:

- (a) 2^{12}
- (b) 2^6
- (c) 0
- (d) 4

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

Un numero naturale è palindromo se letto in senso inverso è identico a sé stesso; ad esempio, 151 e 17271 sono numeri palindromi. Un numero naturale n si dice palizero se ha un numero dispari di cifre, è palindromo e la cifra che appare una sola volta al centro è lo 0. Es. 1234567890987654321 è palizero, 3980893 è palizero, 23732, 23400432 e 124421 sono palindromi ma non palizeri.

Si dica quanti sono i numeri palizeri compresi tra 10^3 e 10^5 estremi esclusi, scegliendo una tra le seguenti alternative.

- (a) 10^2
- (b) $10 * 9$
- (c) $10^2 + (10^3) * 2$
- (d) $10 * 9 + 9$

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

Due treni sono sullo stesso binario e viaggiano uno verso l'altro; la distanza iniziale tra di loro è di 300 km; il primo treno viaggia a 80km/h, il secondo a 70km/h. Un velocissimo colibrì, che vola a 120km/h, parte dalla locomotiva del primo treno e arriva a toccare la locomotiva del secondo, a quel punto si gira e torna indietro fino a toccare la locomotiva del primo, dove si gira e torna indietro e così via finché i due treni si scontrano.

Quanti chilometri ha percorso, complessivamente, il colibrì?

- (a) 160
- (b) 200
- (c) 240
- (d) 300

ESERCIZI DI PROGRAMMAZIONE

Lo pseudocodice è un linguaggio di progettazione per modellare e dettagliare algoritmi prescindendo da qualsiasi linguaggio di programmazione, ma al tempo stesso sottolineando con sufficiente chiarezza ogni aspetto che preme comprendere compiutamente in quel particolare contesto.

La sintassi ammessa è quella che segue:

Pseudocodice	Significato
variable a: integer variable w: integer [14]	dichiarazione della variabile a di tipo intero dichiarazione della variabile w di tipo array di interi fatto da 14 elementi, indicizzati (come nel linguaggio C) da w[0] a w[13]
variable mat: char [10] [10]	dichiarazione della variabile mat di tipo matrice di caratteri di dimensione 10 × 10, indicizzati (come nel linguaggio C) da mat[0][0] a mat[9][9]
a ← 3 a ← a + 1	assegnamento del valore 3 ad a, incremento del valore di a
output a	istruzione per scrivere dati sullo schermo
output "ciao" output "ciao", var	scrive la stringa ciao a video scrive ciao e poi il contenuto della variabile var
array[7] ← 0 mat[5][28] ← 1	esempi di inizializzazione di un elemento di un array e di una cella di una matrice
+ - × / MOD	questi simboli indicano rispettivamente le operazioni aritmetiche di somma, differenza, prodotto, divisione e resto della divisione intera
= < > ≤ ≥ ≠	sono i simboli che indicano i sei tipi di confronto: uguale, minore, maggiore, minore o uguale, maggiore o uguale, diverso
(a ≤ 3) or (a ≥ 5) (a ≤ 3) and (a ≤ 5) !(a)	esempi di operazioni logiche: ! per il NOT logico, and per l'AND, or per l'OR

Attenzione! Queste definizioni non sono un "contratto vincolante". Il Comitato Olimpico potrebbe ritenere utile usare notazioni speciali ai fini della chiarezza e facilità di lettura dell'esercizio. Per esempio:

(a ≤ 3) or (b è un numero primo)

Esempi di strutture di controllo (equivalenti di `if`, `while`, etc ...)

Pseudocodice	Significato
<pre>if condizione then ... end if</pre>	è un costrutto condizionale
<pre>while condizione do ... end while</pre>	è un ciclo a condizione iniziale (pre-posta), ovvero il corpo viene eseguito ogni volta che condizione è VERA
<pre>do ... while condizione</pre>	è un ciclo a condizione finale (post-posta), ovvero il corpo viene eseguito fino a che condizione diventa FALSA
<pre>function funzione(p1:tipo, ..., pn:tipo):tipo ... return valore_restituito end function</pre>	sintassi di una funzione
<pre>procedure nome_procedura(p1:tipo, ..., pn:tipo) ... return end procedure</pre>	sintassi di una procedura (funzione che non restituisce un valore)

I seguenti esercizi servono per verificare le capacità di programmazione usando la sintassi sopra descritta oppure la sintassi del linguaggio C.

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

Data la seguente funzione:

Codice C/C++	Pseudocodice
<pre>1 int aut(int par1, int par2, int par3) { 2 if (par1 == 0) { 3 if (par2 < par3) 4 return 1; 5 else 6 return 0; 7 } else { 8 if (par2 == 0) { 9 if (par1 < par3) 10 return 1; 11 else 12 return 0; 13 } else { 14 if ((sqrt(par1 * par1 + par2 * par2)) < par3) 15 return 1; 16 else 17 return 0; 18 } 19 } 20 }</pre>	<pre>1: function AUT(par1: integer, par2: integer, par3: integer) → integer 2: if par1 = 0 then 3: if par2 < par3 then 4: return 1 5: else 6: return 0 7: end if 8: else 9: if par2 = 0 then 10: if par1 < par3 then 11: return 1 12: else 13: return 0 14: end if 15: else 16: if Sqrt(par1 × par1 + par2 × par2) < par3 then 17: return 1 18: else 19: return 0 20: end if 21: end if 22: end if 23: end function</pre>

Indicare l'affermazione vera tra le seguenti:

- (a) Questa funzione ritorna 1 solo se la coppia $(par1, par2)$ si trova nel cerchio di raggio $par3$
- (b) Questa funzione ritorna 1 se chiamata sulla terna $(1,0,3)$
- (c) Questa funzione ritorna 0 per tutte le terne della forma $(a,2*a, 3*a)$
- (d) Questa funzione deve essere chiamata con $par3 \geq 0$

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

Dato il seguente programma:

Codice C/C++	Pseudocodice
<pre>1 void proc(float *v, int n) { 2 int i = 0; 3 while (i < n) { 4 float b = v[i]; 5 float c = 0; 6 while (b > 1) { 7 int a = 1; 8 while (b > a) { 9 a = a * 2; 10 } 11 if (b < a) { 12 b = b - a / 2; 13 c = c + a / 2; 14 } 15 } 16 if (i % 2 == 0) 17 v[i] = c; 18 else 19 v[i] = c + 1; 20 i = i + 1; 21 } 22 }</pre>	<pre>1: procedure PROC(v: array, n: integer) 2: variable i: integer 3: variable a: integer 4: variable b: float 5: variable c: float 6: i ← 0 7: while i < n do 8: b ← v[i] 9: c ← 0 10: while b > 1 do 11: a ← 1 12: while b > a do 13: a ← a × 2 14: end while 15: if b < a then 16: b ← b - a / 2 17: c ← c + a / 2 18: end if 19: end while 20: if i MOD 2 = 0 then 21: v[i] ← c 22: else 23: v[i] ← c + 1 24: end if 25: i ← i + 1 26: end while 27: end procedure</pre>

Come cambia il vettore $v=(3.45, 5.67, 8.92, 2.12, 7.33, 8.21, 4.21, 9.03)$ quando viene dato in input alla procedura insieme all'intero $n=8$?

- (a) [3.0, 6.0, 8.0, 3.0, 7.0, 9.0, 4.0, 10.0]
- (b) [3.0, 6.0, 7.0, 3.0, 8.0, 9.0, 5.0, 10.0]
- (c) [3.0, 6.0, 7.0, 3.0, 9.0, 9.0, 6.0, 10.0]
- (d) [3.0, 6.0, 8.0, 3.0, 10.0, 9.0, 7.0, 10.0]

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

Cosa scrive a video il seguente programma?

Codice C/C++	Pseudocodice
<pre>1 void pp(float *v, int dim, int k) { 2 int i = 0; 3 while (i < k) { 4 v[dim - i - 1] = 0; 5 i = i + 1; 6 } 7 } 8 9 int main() { 10 float vet[10]; 11 int i = 0; 12 while (i < 10) { 13 vet[i] = (float)25 / 4; 14 i = i + 1; 15 } 16 pp(vet, 10, 5); 17 printf("Lo array dopo l'esecuzione di pp vale: "); 18 i = 0; 19 while (i < 10) { 20 printf("%f ", vet[i]); 21 i = i + 1; 22 } 23 }</pre>	<pre>1: procedure PP(v: float[], dim: integer, k: integer) 2: variable i: integer 3: i ← 0 4: while i < k do 5: v[dim - i - 1] ← 0 6: i ← i + 1 7: end while 8: end procedure 9: variable i: integer 10: variable vet: float[10] 11: i ← 0 12: while i < 10 do 13: vet[i] ← 25 / 4 14: i ← i + 1 15: end while 16: PP(vet, 10, 5) 17: output 'Lo array dopo l'esecuzione di pp vale: ' 18: i ← 0 19: while i < 10 do 20: output vet[i] 21: i ← i + 1 22: end while</pre>

- (a) Lo array dopo l'esecuzione di pp vale 0 0 0 0 0 0 0 0 0 0
- (b) Lo array dopo l'esecuzione di pp vale 6.25 6.25 6.25 6.25 6.25 6.25 6.25 0 0 0
- (c) Lo array dopo l'esecuzione di pp vale 6.25 6.25 6.25 6.25 6.25 0 0 0 0 0
- (d) Lo array dopo l'esecuzione di pp vale 0 0 0 0 0 6.25 6.25 6.25 6.25 6.25

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

Si consideri la seguente procedura.

Codice C/C++	Pseudocodice

Quale disposizione di asterischi viene stampata a schermo tra le seguenti?

<p>a)</p> <pre> * * * * * * * * * </pre>	<p>b)</p> <pre> * * * * * * * * * </pre>
<p>c)</p> <pre> * * * * * * * * * * * </pre>	<p>d)</p> <pre> * * * * * * * </pre>

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

Dato il seguente programma:

Codice C/C++	Pseudocodice
<pre> 1 int main() { 2 float c = 0; 3 int v[] = {1, 7, 4, 9, 10, 32, 56, 22, 34, 6}; 4 int w[] = {5, 65, 87, 43, 32, 87, 5, 3, 1, 0, 8, 6, 4, 2}; 5 int i = 0; 6 while (i < 10) { 7 int j = 0; 8 while (j < 14) { 9 if (v[i] == w[j]) { 10 c = c + 1; 11 } 12 j = j + 1; 13 } 14 i = i + 1; 15 } 16 float p1 = c / 10 * 100; 17 float p2 = c / 14 * 100; 18 printf("p1 = %2.2f, p2 = %2.2f", p1, p2); 19 return 0; 20 } </pre>	<pre> 1: variable i: integer 2: variable j: integer 3: variable v: integer[10] 4: variable w: integer[14] 5: variable c: float 6: c ← 0 7: v ← [1, 7, 4, 9, 10, 32, 56, 22, 34, 6] 8: w ← [5, 65, 87, 43, 32, 87, 5, 3, 1, 0, 8, 6, 4, 2] 9: i ← 0 10: while i < 10 do 11: j ← 0 12: while j < 14 do 13: if v[i] = w[j] then 14: c ← c + 1 15: end if 16: j ← j + 1 17: end while 18: i ← i + 1 19: end while 20: variable p1: float 21: variable p2: float 22: p1 ← c / 10 × 100 23: p2 ← c / 14 × 100 24: SCRIVI('p1 = ', p1, ' p2 = ', p2) </pre>

Scegliere quella corretta fra le seguenti opzioni:

- (a) Il programma stampa $p1 = 40.00$, $p2 = 35.71$
- (b) Il programma stampa il rapporto, in percentuale, degli elementi appartenenti all'intersezione dei due insiemi (v e w) sul totale degli elementi rispettivamente del primo e del secondo insieme
- (c) Il programma stampa il rapporto, in percentuale, degli elementi appartenenti all'unione dei due insiemi sul totale degli elementi rispettivamente del primo (v) e del secondo insieme (w)
- (d) Il programma stampa $p1 = 200.00$, $p2 = 120.71$

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

Date le seguenti funzioni ricorsive:

Codice C/C++	Pseudocodice
<pre>1 int rec1 (int a, int b) { 2 int c = a * b; 3 if (c == 0) { 4 return 0; 5 } 6 return rec2(a, c); 7 } 8 9 int rec2 (int a, int b) { 10 int d = b - 3 * a; 11 if (d % 27 == 0) { 12 return 0; 13 } 14 return rec1(b, d); 15 }</pre>	<pre>1: function REC1(a: integer, b: integer) → integer 2: variable c: integer 3: c ← a × b 4: if c = 0 then 5: return 0 6: end if 7: return REC2(a, c) 8: end function 9: function REC2(a: integer, b: integer) → integer 10: variable d: integer 11: d ← b - 3 × a 12: if d MOD 27 = 0 then 13: return 0 14: end if 15: return REC1(b, d) 16: end function</pre>

Si supponga di eseguire `rec1(4,6)`.

Quante chiamate rispettivamente di `rec1` (**NUMREC1**) e `rec2` (**NUMREC2**) sono necessarie prima che una delle due funzioni restituisca il valore zero? Nelle chiamate di `rec1` contare anche la chiamata iniziale `rec1(4,6)`.

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

Un file di testo conteneva due diverse funzioni per il calcolo del numero di modi diversi in cui è possibile ordinare n oggetti.

Le linee dello pseudocodice sono state accidentalmente disordinate, come riportato qui sotto.

Codice C/C++	Pseudocodice
<pre> A: if (n == 1) { B: if (n == 1) { C: return 1; D: } E: int f2(int n) { F: } G: int f1(int n) { H: return n * f1(n - 1); I: risp = 0; L: } M: return 1; N: int risp; O: int i; P: while (i <= n) { Q: risp = risp + f2(n - 1); R: i = 1; S: i = i + 1; T: } U: return risp; V: }</pre>	<pre> A: if n = 1 then B: if n = 1 then C: return 1 D: end if E: function F2(n: integer) → integer F: end function G: function F1(n: integer) → integer H: return n × F1(n - 1) I: risp ← 0 L: end if M: return 1 N: variable risp: integer O: variable i: integer P: while i ≤ n do Q: risp ← risp + F2(n - 1) R: i ← 1 S: i ← i + 1 T: end while U: return risp V: end function</pre>

Per ricostruire l'ordinamento corretto delle linee di codice è necessario produrre una lista di coppie (LETT, NUM), in cui LETT corrisponde alla lettera che identifica una riga, come indicato sopra e NUM corrisponde al numero che tale riga avrebbe nell'ordinamento corretto (nota: i numeri di riga partono da 1), come indicato nella seguente griglia, in cui alcuni numeri sono già stati inseriti:

A	B	C	D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V
	2	3	4		6			9	12	11		13		16			18		20

Si chiede quindi di riportare i numeri corretti associati alle lettere **A, E, G, H, N, P, R, S** e **U** che completino correttamente la suddetta griglia.

ESERCIZI A CARATTERE ALGORITMICO

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

Nell'informatica si parla di "edit distance" quando si vuole misurare quanto sono diverse due parole w_1 e w_2 . Si dice che due parole w_1 e w_2 hanno distanza:

- 1 se w_1 è ottenuta da w_2 modificando una lettera (ad esempio, sono a distanza 1 "cane" e "cene") o viceversa;
- 2 se w_1 è ottenuta da w_2 inserendo una lettera in una qualunque posizione (ad esempio, sono a distanza 2 "mangia" e "mangiai") o viceversa.

Luca ha saputo che Mario partecipa alle Olimpiadi di Informatica e ha deciso di cercare su Google che cosa sono, ma ha commesso alcuni errori di battitura e ha scritto: "Olinpiadi Italianer de Informatia".

Sapendo che la distanza tra due frasi è la somma delle distanze tra le parole corrispondenti, indicare **DIST**, ovvero quanto la frase scritta da Luca si discosta da "Olimpiadi Italiane di Informatica".

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

Siano A e B due insiemi tali che $A = \{1, 2, 5, 8\}$ e $B = \{3, 5, 9, 11, 42\}$.

Si definisce $D(x, X)$ il numero di elementi presenti in X di cui x è un divisore (formalmente $D(x, X) = \#\{y \text{ in } X \text{ such that } x \mid y\}$).

Indicare il più piccolo numero **c** tale che risulti $D(c, A) > D(c, B)$.

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

Data la funzione $f(x) = 2x \pmod{7}$ (ovvero $f(x)$ è il resto ottenuto dividendo $2x$ per 7) si consideri la seguente tabella, denominata **BF1**:

Posizione	0	1	2	3	4	5	6
Valore	0	0	0	1	0	0	1

Una generica tabella di tipo BF rappresenta un insieme di interi.

La regola per inserire valori nella tabella BF è la seguente: inizialmente sono tutti zero. Se si vuole inserire in BF un intero x , si deve applicare a x la funzione f e poi scrivere un 1 nella posizione numero $f(x)$. Se era già presente un 1 in posizione $f(x)$ non si deve fare niente.

Ad esempio, se si vuole inserire nella tabella **BF1** il numero 6 non si deve far altro che osservare che $6 \cdot 2$ modulo 7 fa 5 e inserire un 1 in posizione 5, ottenendo la tabella **BF2**:

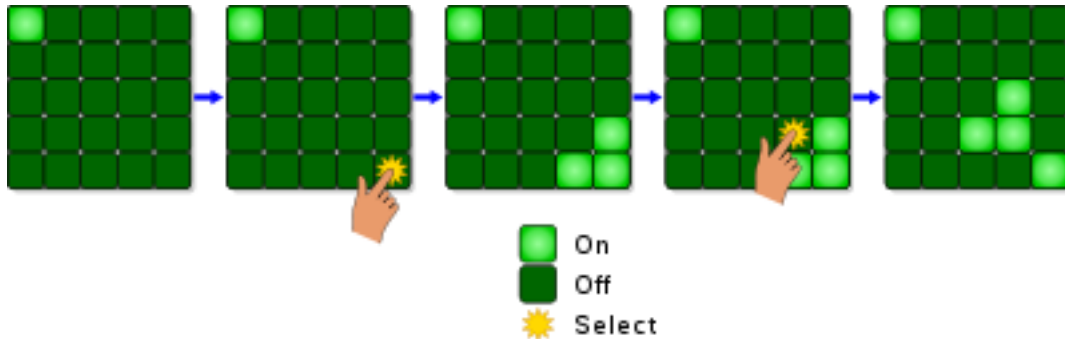
Posizione	0	1	2	3	4	5	6
Valore	0	0	0	1	0	1	1

Indicare la corretta fra le seguenti affermazioni, riferite alla prima tabella **BF1**:

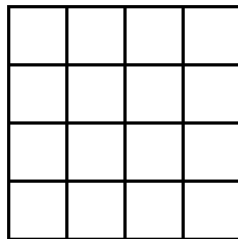
- In **BF1** è presente il numero 4
- In **BF1** non è presente il numero 19
- In **BF1** potrebbe essere presente il numero 6
- In **BF1** potrebbe essere presente il numero 12

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

Nel gioco **Lights Out** si ha una matrice di 5x5 luci, che possono essere accese o spente. Premendo su un elemento della matrice, si cambia lo stato di quell'elemento e dei suoi quattro vicini (alto, basso, destra e sinistra), come mostrato nella figura di seguito.



Si consideri una versione semplificata, con una matrice 4x4 come quella mostrata nella figura qui sotto.



Inizialmente le luci sono tutte spente. Una mossa consiste nel premere un elemento della matrice. Qual è il numero minimo **NUMMOSSE** che bisogna fare per arrivare alla configurazione in cui tutte le luci sono accese?

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

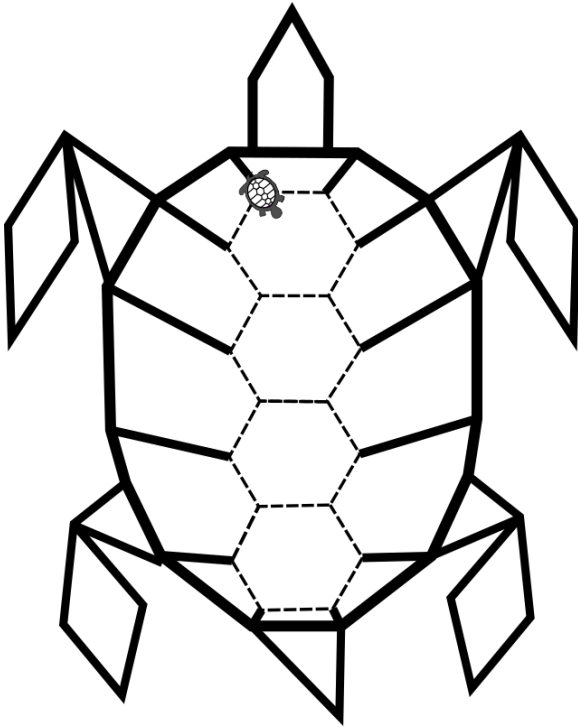
Si prenda $R = \{00101, 101, 1010101, 1111001\}$.

Si indichi una stringa binaria w (fatta di soli 0 e 1) che contenga al più 13 caratteri tale che ogni stringa presente nell'insieme R sia una sottostringa di w .

Esercizio N° 18 – La risposta esatta vale 2 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 sotto la 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.



Quest'anno la tartaruga vuole realizzare un autoritratto. Prima ha disegnato la sua sagoma a matita (in figura la linea tratteggiata) e poi ha iniziato a ricalcarla con la penna.

In questo momento la tartaruga si trova nel vertice in alto a sinistra dell'esagono più alto ed è nella condizione `pennagiu`, sapendo che gli esagoni del carapace sono regolari e hanno lato 1, scegliere tra le seguenti quattro alternative quella che **non** fa il disegno corretto.

(Alternativa A)

```
[1] ripeti 4 {
[2]   destra 60
[3]   avanti 1
[4]   sinistra 60
[5]   avanti 1
[6] }
[7] ripeti 4 {
[8]   sinistra 60
[9]   avanti 1
[10] }
[11] ripeti 3 {
[12]   pennasu
[13]   destra 180
[14]   avanti 1
[15]   pennagiu
[16]   ripeti 3 {
[17]     sinistra 60
[18]     avanti 1
[19]   }
[20] }
```

(Alternativa B)

```
[1] ripeti 4 {
[2]   destra 60
[3]   avanti 1
[4]   sinistra 60
[5]   avanti 1
[6] }
[7] ripeti 4 {
[8]   sinistra 60
[9]   avanti 1
[10] }
[11] ripeti 3 {
[12]   pennasu
[13]   sinistra 180
[14]   avanti 1
[15]   pennagiu
[16]   ripeti 3 {
[17]     sinistra 60
[18]     avanti 1
[19]   }
[20] }
```


(Alternativa C)

```

[1] ripeti 4 {
[2]   avanti 1
[3]   destra 60
[4]   sinistra 60
[5]   avanti 1
[6] }
[7] ripeti 4 {
[8]   sinistra 60
[9]   avanti 1
[10] }
[11] ripeti 3 {
[12]   pennasu
[13]   destra 180
[14]   avanti 1
[15]   pennagiu
[16]   ripeti 3 {
[17]     sinistra 60
[18]     avanti 1
[19]   }
[20] }

```

(Alternativa D)

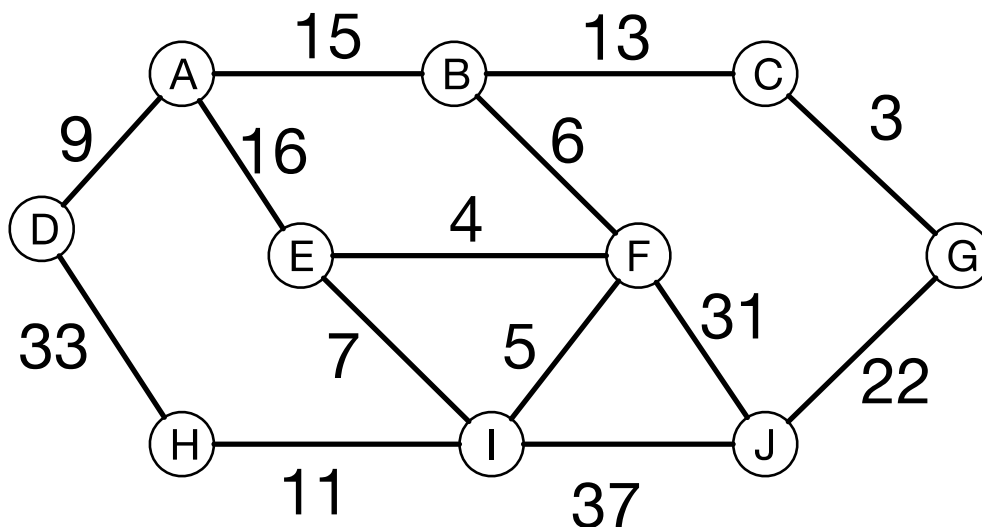
```

[1] ripeti 4 {
[2]   destra 60
[3]   avanti 1
[4]   sinistra 60
[5]   avanti 1
[6] }
[7] ripeti 4 {
[8]   sinistra 60
[9]   avanti 1
[10] }
[11] ripeti 3 {
[12]   destra 180
[13]   pennasu
[14]   avanti 1
[15]   pennagiu
[16]   ripeti 3 {
[17]     sinistra 60
[18]     avanti 1
[19]   }
[20] }

```

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

In un vecchio edificio ci sono 10 computer che devono essere collegati in rete. Dato l'elevato spessore delle pareti, non è possibile usare una rete wifi e si decide, quindi, di collegarli via cavo. Non sono possibili tutti i collegamenti, e ogni collegamento ha un costo diverso. Si deve aiutare a progettare la rete scegliendo i nove collegamenti necessari per fare in modo che ogni computer sia collegato alla rete (ovvero ad almeno un altro computer) e che il costo complessivo sia minimo. Dopo che i nove collegamenti saranno stati scelti, indicare il costo totale **TOTCOSTO**, pari alla somma dei costi dei nove collegamenti selezionati.









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

La famosa Sushi Squad, composta da 5 studenti delle scuole superiori, si trova nel noto sushi bar Minimax Hao. Dal tavolo del sashimi (riportato in figura) i 5 protagonisti possono prendere un solo piatto, ma la squadra si divide in 2: gli amanti del salmone (S) e gli amanti del tonno (T), in cui ognuna delle due fazioni vuole che dal tavolo venga preso il piatto contenente la maggior quantità del pesce preferito.

Le due squadre S e T si disputano la scelta del piatto giocando secondo questa logica: la squadra S ha diritto a selezionare una riga, la squadra T una colonna.

Valutare i due seguenti scenari.

	C1	C2
R1	 3 Salmone 1 Tonno	 2 Salmone 3 Tonno
R2	 5 Salmone 0 Tonno	 1 Salmone 4 Tonno
R3	 1 Salmone 2 Tonno	 4 Salmone 4 Tonno

A) Sia la squadra **S** ad **iniziare** il gioco.

S seleziona la **riga** in modo tale che, qualunque sia la colonna che sceglierà T nella mossa successiva, sia massimo il numero minimo di filetti di salmone nel piatto. T, quando arriva il suo turno, sceglie semplicemente il piatto con più filetti di tonno.

B) Sia la squadra **T** ad **iniziare** il gioco.

T seleziona la **colonna** in modo tale che, qualunque sia la riga scelta da S nella mossa successiva, sia massimo il numero minimo di filetti di tonno nel piatto. Quando arriva il turno di S, questa sceglie semplicemente il piatto con più filetti di salmone.

Si devono indicare **RA** e **CA** (ovvero i numeri di riga e colonna scelti nello scenario A) e **RB** e **CB** (ovvero i numeri di riga e colonna scelti nello scenario B). Per esempio, la risposta "1" (numero intero di una sola cifra) per RA indica che si intende dare come risposta la prima riga dello scenario A.