

- 1) La codifica binaria in modulo e segno di +11
a) è sempre lunga esattamente 5 bit
b) è sempre lunga esattamente come la codifica in complemento a 2 di +11
c) ha una lunghezza che può variare (a seconda dei bit a disposizione) **X**
d) ha una lunghezza che può variare da un minimo di 5 bit a un massimo di 10
- 2) La parola chiave VOID in C++
a) indica sempre un tipo di dato
b) non indica mai un tipo di dato **X**
c) a volte indica un tipo di dato
d) spesso indica un tipo di dato
- 3) Nel codice WHILE ((x != ____)&&(x > ____)) {cout << x; x++;} ci sono spazi da riempire.
a) La cout non viene mai eseguita se gli spazi vengono riempiti con uno 0
b) Esiste un modo di riempire gli spazi che non fa mai eseguire la cout **X (ad es. x!=x && x > x)**
c) La cout viene sempre eseguita almeno una volta se gli spazi vengono riempiti con un 1
d) Non esiste un modo di riempire gli spazi per far andare il codice in loop **esiste: x!= x-1 && x> x-1**
- 4) L'espressione logica (A||!A)&&(A&&!A)
a) ha lo stesso valore di A
b) ha lo stesso valore di !A
c) ha lo stesso valore di A&&!A **X (sempre false entrambi)**
d) ha lo stesso valore di A||!A
- 5) Se la scheda SD ha capacità di 4GB e una variabile di tipo int occupa 32bit, la scheda può contenere
a) circa 1 milione di variabili int
b) circa 1 miliardo di variabili int **X**
c) circa 100 miliardi di variabili int
d) circa 1000 miliardi di variabili int
- 6) La codifica in complemento a 2 di -199 è
a) 110111001 b) 100110001 c) 101011001 d) 100111001 **X**
- 7) Nei diagrammi di flusso
a) non esistono elementi da cui partono 2 frecce
b) esistono elementi da cui non parte nessuna freccia **X (l'elemento che indica la fine dell'algoritmo)**
c) non esistono elementi in cui arrivano 2 frecce
d) non esistono elementi a cui non arriva nessuna freccia
- 8) La memoria RAM
a) è dove si trovano i valori delle variabili di un programma in esecuzione **X**
b) conserva il suo contenuto anche dopo lo spegnimento del calcolatore
c) è la componente del calcolatore dove si eseguono le operazioni aritmetiche
d) è al livello più basso della gerarchia di memorie
- 9) Un array dichiarato così: char parola[10];
a) è di un tipo user-defined e semplice
b) è di un tipo user-defined e strutturato
c) è di un tipo built-in e semplice
d) è di un tipo built-in e strutturato **X**
- 10) Sono tutte periferiche di output:
a) il monitor, il touch-pad, e la stampante
b) il monitor, la stampante, e il disco fisso
c) il proiettore, il monitor, e la stampante **X**
d) il monitor, la stampante, la tastiera
- 11) Dopo l'istruzione k+= 0;
a) il valore di k è stato modificato
b) k vale sicuramente 0
c) k può valere 0 **X (se valeva 0 già da prima)**
d) k non può valere 1 **k può valere 1 (se valeva 1 già da prima)**

12) 100 è la codifica binaria in modulo e segno del seguente numero:

- a) 100 b) 0 **X** c) 4 d) -4

13) L'acronimo RAM significa

- a) Random Array Memory
b) Random Access Matrix
c) Random Array Matrix
d) Random Access Memory **X**

14) Per ogni problema

- a) esiste una soluzione che lo risolve
b) se esiste una soluzione, esiste un algoritmo che lo risolve
c) se esiste un algoritmo scritto in C++ che lo risolve, esiste un programma che lo risolve **X**
d) esiste un algoritmo che lo risolve

15) Nell'algoritmo ricorsivo per il calcolo del fattoriale del numero intero n

- a) n vale 10 nel caso base
b) n vale 0 nell'ipotesi ricorsiva
c) la soluzione finale è data dall'ipotesi ricorsiva
d) il passo consiste in un'operazione di moltiplicazione **X**

16) Per avere un loop nel ciclo FOR (int i = 0; i ____; i++) cout << i; nello spazio scriviamo

- a) <100 b) >100 c) > -100 **X** d) < -100

17) In una procedura

- a) i parametri non vengono mai modificati
b) non è presente un'istruzione di return **X**
c) il tipo del parametro in uscita può essere int
d) i parametri vengono sempre passati per indirizzo

18) Data una variabile di un tipo definito tramite STRUCT,

- a) si accede ai suoi campi con la dot notation **X**
b) si accede al suo valore con la dot notation
c) essa non può fungere da parametro di una funzione
d) essa non può a sua volta contenere campi strutturati

19) Per costruire un circuito elettronico che dà sempre 0 in output

- a) serve un segnale in ingresso, una porta NOT, e una porta OR
b) serve un segnale in ingresso, una porta NOT, e una porta AND **X** (si sdoppia l'ingresso, una copia viene negata, e tutto finisce in AND per ottenere A AND !A che è sempre 0)
c) servono due segnali in ingresso e una porta OR
d) servono due segnali in ingresso e una porta AND

20) Nel codice IF (cond1) istr1; ELSE IF (cond2) istr2;

- a) istr2 viene eseguita tutte le volte in cui cond1 è falsa
b) istr2 viene eseguita tutte le volte in cui cond2 è vera
c) istr2 viene eseguita tutte le volte in cui (cond1 && cond2) è falsa
d) istr2 viene eseguita tutte le volte in cui cond1 è falsa e cond2 è vera **X**

Programmazione 1)

Scrivere una funzione QUAD che prende in input un intero, e restituisce un booleano: TRUE se il numero è un quadrato perfetto, FALSE altrimenti.

```
bool quad(int n){
    for (int i=0; i<=n/2; i++)
        if (i*i == n)
            return true;
    return false;
}
```

Programmazione 2)

Scrivere una funzione TRI che prende in input una matrice quadrata di interi e la sua dimensione, e restituisce un booleano: TRUE se la matrice è triangolare, FALSE altrimenti. Si supponga che il parametro in input sia sempre una matrice quadrata. Una matrice quadrata si dice triangolare se tutti i valori sotto la diagonale principale sono 0. Ad es:

1	2	3		1	2	3
---	---	---	--	---	---	---

0 4 5 è diagonale, mentre 0 4 5 non lo è a causa dell'1 nella posizione (3,1).

0	0	6		1	0	6
---	---	---	--	---	---	---

```
bool tri(int m[], int dim){
    for (int i=1; i<dim; i++)
        for (int j=0; j < i; j++)
            if (m[i][j] != 0)
                return false;
    return true;
}
```