

- 1) In una funzione
  - a) a volte non è compresa l'istruzione return
  - b) il codice viene sempre compilato senza errori
  - c) viene sempre dichiarato il tipo del valore restituito **X**
  - d) nessuna delle precedenti
- 2) Una variabile così dichiarata: float x = 1.5;
  - a) dopo un'istruzione x = y nel codice ha un valore diverso da 1.5
  - b) potrebbe mantenere lo stesso valore fino alla fine del programma **X**
  - c) non può assumere valori numerici interi
  - d) non può essere usata come parametro attuale di una procedura
- 3) L'espressione !(Z&&!Z)&&(!Z&&Z)
  - a) non è corretta secondo le regole del C++
  - b) ha sempre valore true
  - c) ha un valore che dipende dal valore di Z
  - d) nessuna delle precedenti **X**
- 4) In un ciclo for
  - a) tutte le istruzioni vengono eseguite almeno 3 volte
  - b) c'è sempre una variabile che viene incrementata di 1
  - c) c'è sempre una variabile che viene decrementata di 1
  - d) nessuna delle precedenti **X**
- 5) Per la codifica binaria in modulo e segno di -88
  - a) servono 7 bit
  - b) servono 8 bit **X**
  - c) servono 9 bit
  - d) nessuna delle precedenti
- 6) Un dispositivo nel calcolatore che è solo di output è
  - a) la tastiera
  - b) il monitor **X**
  - c) il mouse
  - d) la memoria centrale
- 7) Dopo l'istruzione k/= 2;
  - a) k diventa 1 se prima valeva 2 **X**
  - b) dà errore se k vale 0
  - c) k viene decrementato di 2
  - d) nessuna delle precedenti
- 8) Nel seguente codice: if (A||B) istr1; else istr2;
  - a) la falsità di B è sufficiente a garantire l'esecuzione di istr2
  - b) la verità di A è sufficiente a garantire la mancata esecuzione di istr2 **X**
  - c) se sia A sia B sono false, istr2 non viene eseguita
  - d) nessuna delle precedenti
- 9) La corretta compilazione di un programma scritto in C++
  - a) è garanzia del fatto che una soluzione al problema da risolvere è stata trovata
  - b) è necessaria alla soluzione del problema, ma non ne è una garanzia **X**
  - c) è sempre garantita
  - d) nessuna delle precedenti
- 10) La stringa in codice binario 1111000 rappresenta
  - a) il numero 23 in modulo e segno
  - b) il numero 75 in complemento a 2
  - c) il numero -89 in complemento a 2
  - d) nessuna delle precedenti **X**
- 11) Nell'algoritmo ricorsivo per controllare che una stringa sia palindroma (si legge allo stesso modo da destra e da sinistra), il caso base è
  - a) la stringa è vuota o ha un solo carattere **X**
  - b) la stringa è palindroma
  - c) la stringa non è palindroma
  - d) nessuna delle precedenti

12) Un componente elettronico che esegue la funzione logica NOT si rappresenta graficamente con:

- a) un semicerchio
- b) un triangolo seguito da un piccolo cerchio **X**
- c) una forma simile a una falce di luna
- d) un triangolo preceduto da un piccolo cerchio

13) Il ciclo `for(i=100; i >= K; i--) { cout << i; }`

- a) non termina mai
- b) non termina se K è nullo
- c) termina con la stampa su schermo del valore di K
- d) nessuna delle precedenti **X (l'ultima cosa che il ciclo FOR fa è il confronto tra i e K, quando i vale K-1)**

14) Se x è un float che vale 1.5 e y è un ~~int~~ float che vale 0.6, dopo aver eseguito `z = x + y`; z vale

- a) 2.1 se z è un int
- b) 2 in ogni caso
- c) 2.1 se z non è un float
- d) 2 se z è un int **X**

**La domanda presenta un errore nel testo, motivo per cui ogni risposta verrà considerata corretta. Con il testo corretto, la risposta corretta è la d.**

15) Un algoritmo

- a) è un programma
- b) a volte non può essere trasformato in un programma
- c) è un programma se scritto in inglese
- d) non è un programma **X**

16) Quanti bit sono contenuti in una RAM da 4GB (gigabyte) ?

- a)  $2^{32}$
- b)  $2^{33}$
- c)  $2^{35}$  **X**
- d)  $2^{36}$

17) Il passaggio di parametri per indirizzo

- a) si può usare solo nelle procedure perché le funzioni lavorano solo con le copie dei parametri attuali
- b) si rende necessario quando si vogliono salvaguardare i valori originali dei parametri attuali
- c) comporta un risparmio di memoria perché non vengono fatte copie dei parametri attuali **X**
- d) nessuna delle precedenti

18) In un diagramma di flusso:

- a) non esiste mai un rombo
- b) se c'è un rombo allora nel programma corrispondente ci sarà sicuramente un costrutto IF
- c) se c'è un rombo allora nel programma corrispondente ci sarà sicuramente un costrutto WHILE
- d) nessuna delle precedenti **X**

19) Un tipo di dato strutturato

- a) può contenere diversi campi di tipi diversi, a loro volta strutturati **X**
- b) esiste solo se definito dall'utente
- c) può contenere diversi campi di tipi diversi, ma solo semplici e predefiniti
- d) nessuna delle precedenti

20) Dati 1111 e 0001 in complemento a 2, la loro somma:

- a) dà zero senza overflow **X**
- b) dà zero con overflow
- c) dà -16 senza overflow
- d) dà -16 con overflow

Programmazione 1)

Scrivere il codice della procedura "passabasso" che, dati in ingresso un array di interi, la sua dimensione, e un valore intero di soglia, stampa su schermo i valori dell'array se sono strettamente inferiori al valore di soglia, e una X maiuscola al posto di ogni valore uguale o sopra la soglia. Ad esempio, "passabasso" applicato a ([33 4 -3 44 8 12], 6, 10) deve fare visualizzare su schermo quanto segue: X 4 -3 X 8 X

**cfr. soluzione dell'appello di Aprile, dove si trova un esercizio molto simile**

Programmazione 2)

Scrivere il codice della procedura "mischia" che, dati in ingresso due array di caratteri e due interi che rappresentano le dimensioni degli array, stampa su schermo i caratteri dei due array alternati.

Ad esempio, "mischia" applicato a ("palla", 5, "boh", 3) deve stampare su schermo la stringa "p b a o l h l a". Da notare che una volta finite le lettere di un array, si stampano solo le lettere rimanenti dell'altro.

una possibile implementazione è:

```
void mischia(char v1[], int dim1, char v2[], int dim2){
    int min, i;
    if (dim1 >= dim2)
        min = dim2;
    else
        min = dim1;
    for (i=0; i < min; i++)
        cout << v1[i] << " " << v2[i] << " ";
    if (min==dim1)
        for(i=min; i < dim2; i++)
            cout << v2[i] << " ";
    else
        for(i=min; i < dim1; i++)
            cout << v1[i] << " ";
}
```