

Fondamenti di Informatica
Ingegneria Clinica
Università di Roma “La Sapienza”
Prova del 27/07/2009

Le risposte corrette valgono 1 punto, quelle errate -0,25 punti, quelle non date 0 punti
Nota: Tra le proposte per ciascuna domanda una sola è esatta

Nome: **Cognome:**

Matricola:

Aula: **Fila:** **Posizione:**

1. Si consideri l'architettura di un Personal Computer.

A Il registro PC (Program Counter) contiene informazioni riguardanti l'ultima operazione eseguita dalla ALU.

B Ad ogni ciclo il registro Flag (o PSW) contiene l'istruzione da eseguire nello stesso ciclo.

C La memoria centrale può contenere i dati del programma correntemente in esecuzione.

D La memoria di massa è volatile.

2. In merito ai linguaggi di programmazione quali delle seguenti affermazioni sono vere?

A Un programma scritto in linguaggio **assembler** non necessita di traduzione.

B Il **debugger** serve per trovare più facilmente errori sintattici nei programmi.

C Un compilatore per il linguaggio C è un programma che è indipendente dall'architettura HW/SW del computer utilizzato.

D Un compilatore traduce un programma scritto con un linguaggio di alto livello in uno scritto con un linguaggio a basso livello, ma non viceversa.

3. Si consideri il seguente programma C:

```
#include <stdio.h>
main()
{int x[4]={0, 3, 4, 8};
int j, A, B=0, C;
for(j=3; j>=0; j--) /*istruzione 1*/
switch(x[j]%3-1) /*istruzione 2*/
{case 0: x[j]--;
case 1: A=x[j]; break;
case 2: C=0; break;
default: B--A;
}
C=x[0]+x[2] - B; /* istruzione 3*/
}
```

A Immediatamente dopo l'istruzione 3, la variabile C ha il valore 3.

B Al termine dell'esecuzione, il valore di tutti gli elementi di x risultano modificati rispetto ai loro valori iniziali.

C Al termine dell'esecuzione, il valore di x[3] è uguale a 7.

D Immediatamente dopo il ciclo for, la variabile B ha il valore 1.

4 Si consideri il seguente programma C:

```
#include <stdio.h>
main()
{int A; float B; char C, ch;
A=4; B=-2; C='C'; ch='D';
A=( A%(int)B ?++A: ch-C); /* istruzione 1 */
B=A++/B; /* istruzione 2 */
ch=C+=1; ch-=C; /* istruzione 3 */
}
```

A Immediatamente dopo l'esecuzione dell'istruzione 2, la variabile A ha il valore 2.

B Immediatamente dopo l'esecuzione dell'istruzione 3, la variabile C ha valore 'C'.

C L'operatore di casting nell'istruzione 1 è influente.

D Il programma può generare errori durante l'esecuzione.

5. Si considerino i tipi di dato scalari primitivi nel linguaggio C:

A La sottrazione tra un float e un char produce un errore in compilazione.

B Nel linguaggio C il vettore e' un tipo scalare primitivo.

C L'applicazione della regola di conversione implicita produce una espressione in cui tutti gli operandi assumono uno stesso tipo.

D La conversione esplicita non può trasformare un dato scalare in uno strutturato.

6. Si considerino i vettori nel linguaggio C:

A Un vettore di caratteri può essere considerato una stringa se non contiene il carattere '0'.

B Nella definizione di un vettore, la dimensione può essere espressa mediante il nome di una variabile.

C L'indice di un elemento può essere una variabile di tipo char.

D L'indice di un elemento può essere una variabile di tipo float.

7. Si consideri il seguente programma C:

```
#include <stdio.h>
#define N 9
int f(int *a, int b) {
if( *a < *(a+b) ) /* punto 1 */
return b;
else
return *a=f( a, b+1 )+b;
}
main() {
int i, V[N]={0,1,2,3,4,5,6,7,8};
for ( i=N-1; i>=0; i-=2 )
V[i]=(i+5)/2;
/*punto 2 */
printf("%d\n\n", f(V,0));
/* punto 3 */
for (i=0; i<N; i++) /* punto 4*/
printf("%d\n", V[i]);
}
```

A L'istruzione prima del punto 2 ($V[i] = (i+5)/2$) viene eseguita 4 volte.

B L'istruzione al punto 1 ($\text{if} (*a < *(a+b))$) viene eseguita 3 volte.

C L'istruzione al punto 3 stampa il valore 8.

E Al punto 1 la variabile b vale 7.

8. Si considerino le funzioni nel linguaggio C:

A Tutte le variabili dichiarate nelle definizioni di funzione sono variabili globali.

B I parametri formali vengono allocati nella memoria di massa.

C Un file sorgente C, per essere compilato correttamente, deve contenere tutte le definizioni di tutte le funzioni che utilizza.

D Dall'interfaccia (o prototipo) di una funzione è possibile ricavare il numero e tipo dei parametri da usare per invocarla correttamente.

9. Si considerino le funzioni nel linguaggio C:

A Scegliere dei nomi di funzioni e di parametro significativi renderà i programmi più leggibili ma comporterà un eccessivo uso di commenti per le specifiche.

B Utilizzare gli stessi nomi per gli argomenti passati ad una funzione e per i corrispondenti argomenti inseriti nella definizione è scorretto.

C I programmi dovrebbero essere scritti come collezioni di funzioni piccole. Ciò li renderà più semplici da scrivere, collaudare, modificare.

D Le istanze di funzioni utilizzano dei record di attivazione che vengono deallocati ogni volta che vengono generate ricorsivamente altre istanze della stessa funzione.

10. Si considerino le istruzioni del linguaggio C:

A Ogni istruzione **while** è sempre esprimibile mediante un'istruzione **for**.

B Nell'istruzione **for** l'espressione di modifica viene eseguita nel momento in cui la condizione è falsa.

C Nell'istruzione **do..while** se la condizione è verificata, si esce dal ciclo.

D Nell'istruzione **switch** almeno un blocco di istruzioni è sempre eseguito.