



Rispondere alle domande a risposta multipla annerendo la casella corrispondente alla risposta corretta. Ogni domanda ha una ed una sola risposta corretta.

Cognome e Nome:

Matricola:

Domanda 1 La ricorsione in coda:

- ☐ Permette di risolvere il problema della ricorsione infinita
- ☐ Richiede di non scrivere mai la chiamata ricorsiva come ultimo statement di una subroutine
- ☐ Richiede di non ritornare mai direttamente il valore ritornato da una chiamata ricorsiva
- ☐ Nessuna delle altre risposte
- ☒ Permette di evitare un'eccessiva crescita della dimensione dello stack

Domanda 2 Dato il frammento di programma (espresso in pseudo-codice) contenuto in Figura 1, quanto vale la variabile globale `c` dopo aver eseguito `topolino()`, assumendo scope statico?

- ☐ 14
- ☐ Nessuna delle altre risposte
- ☒ 6
- ☐ 3
- ☐ Non è possibile dirlo

Domanda 3 Dato il frammento di programma (espresso in pseudo-codice) contenuto in Figura 2, qual'è il valore di ritorno di `topolino()`, assumendo scope statico?

- ☐ Nessuna delle altre risposte
- ☐ 0
- ☐ Non è possibile dirlo
- ☒ -3
- ☐ 11

Domanda 4 Se gli array sono memorizzati *per colonne* e `char a[100][100]` è un array multidimensionale di caratteri con `a[0][0]` che ha indirizzo `0x1100`, qual'è l'indirizzo di `a[5][10]`?:

- ☐ Nessuna delle altre risposte
- ☒ 0x24ED
- ☐ 0x21FE
- ☐ 0x22FE
- ☒ 0x14ED



```
int a, b, c;

void pippo(void)
{
    int a;

    a = 6;
    b = 5;
}

void pluto(void)
{
    int c;
    int b;

    pippo();
    c = 3;
    a = 4;
}

void topolino(void)
{
    int a;

    a = 1;
    b = 10;
    pluto();

    c = a + b;
}
```

Figure 1: Esempio di pseudocodice

```
int x, y;

void pippo(void)
{
    x = 8;
    y = 4;
}

void pluto(void)
{
    int y;

    pippo();
    y = 3;
}

int topolino(void)
{
    int x, z;

    x = 5;
    y = 15;
    z = x + y;
    pluto();

    return z - y - x;
}
```

Figure 2: Esempio di pseudocodice



Domanda 5 Un *interprete* di un linguaggio \mathcal{L} scritto in un linguaggio \mathcal{L}_O è:

- ☐ Nessuna delle altre risposte
- ☐ L'implementazione di una macchina astratta scritta nel linguaggio \mathcal{L}_O , che capisce programmi scritti nel linguaggio \mathcal{L}
- ☐ Una implementazione di macchine astratte indipendente dalla macchina fisica
- ☒ Un programma scritto nel linguaggio \mathcal{L}_O che riceve come ingresso un programma $P^{\mathcal{L}}$ (espresso nel linguaggio \mathcal{L}) ed il suo input I generando lo stesso output che genera $P^{\mathcal{L}}$ con input I
- ☐ Un programma che trasforma un programma $P^{\mathcal{L}}$ (espresso nel linguaggio \mathcal{L}) in un programma $P^{\mathcal{L}_O}$ (espresso nel linguaggio \mathcal{L}_O) tale che per ogni input I si ha $P^{\mathcal{L}}(I) = P^{\mathcal{L}_O}(I)$

Domanda 6 β -riducendo $(\lambda a.aab)((\lambda a.aab)(\lambda a.(\lambda b.ba)c))$ si ottiene:

- ☐ La riduzione non termina
- ☐ $ccb(ccb)c$
- ☐ aab
- ☐ $\lambda a.aab$
- ☒ Nessuna delle altre risposte

2. x $\left(((\lambda a.aab)(\lambda a.(\lambda b.ba)c)) \right) b$
 $ccb(ccb)b$

Domanda 7 La *frammentazione esterna* causa:

- ☐ Il funzionamento non corretto di programmi che allocano memoria dinamicamente
- ☐ Un rallentamento rilevante nelle operazioni di allocazione della memoria
- ☐ Nessuna delle altre risposte
- ☒ L'impossibilità di allocare grandi blocchi di memoria anche se la memoria libera totale è sufficiente
- ☐ Uno spreco di memoria

Domanda 8 Il concetto di *variabile modificabile*:

- ☒ E' tipico del paradigma di programmazione imperativo
- ☐ Permette di evitare il fenomeno dell'*aliasing*
- ☐ Nessuna delle altre risposte
- ☐ E' l'unico concetto utilizzabile quando si parla di variabili
- ☐ E' imposto dall'architettura di Von Neumann (variabili non modificabili richiederebbero macchine astratte caratterizzate da memoria a sola lettura)

Domanda 9 Si consideri lo pseudo-codice di Figura 3. Qual'è il valore di ritorno di $f(1, r(1), 1)$ se i parametri sono passati *per nome*?

- ☐ Nessuna delle altre risposte
- ☐ Non è possibile dirlo senza conoscere il tipo di scope (statico o dinamico) utilizzato
- ☒ Si ha ricorsione infinita
- ☐ 1
- ☐ 0



```
int r(int x)
{
    return r(x - 1);
}

int f(int a, int b, int c)
{
    if (c == 1) return a; else return b;
}
```

Figure 3: Esempio di pseudocodice

```
int c = 2;

int pippo(int a)
{
    c = c + 2;
    return a * 2;
}

int pluto(void)
{
    return(pippo(c + 1));
}
```

Figure 4: Esempio di pseudocodice

Domanda 10 Si consideri lo pseudo-codice di Figura 4. Qual'è il valore di ritorno di `pluto()` se i parametri sono passati *per riferimento*?

- ☒ Non è possibile passare `c + 1` per riferimento
- ☐ Dipende dal tipo di scope (statico o dinamico) utilizzato
- ☐ 6
- ☐ Nessuna delle altre risposte
- ☐ 10

Domanda 11 La *frammentazione interna* causa:

- ☐ Il funzionamento non corretto di programmi che allocano memoria dinamicamente
- ☐ Un rallentamento rilevante nelle operazioni di allocazione della memoria
- ☐ Nessuna delle altre risposte
- ☒ Uno spreco di memoria
- ☐ L'impossibilità di allocare grandi blocchi di memoria anche se la memoria libera totale è sufficiente

Domanda 12 β -riducendo $(\lambda a.aaa)((\lambda b.b)(\lambda c.c))$ si ottiene:

- ☒ Nessuna delle altre risposte
- ☐ `aaa`
- ☐ La riduzione non termina
- ☐ `$\lambda x.xa$`
- ☐ `$\lambda a.a$`

2- equiv?