



```
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 1: Esempio di pseudocodice

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 In presenza di variabili modificabili:

- ☐ La valutazione del comando di assegnamento restituisce sempre un valore
- ☐ Esistono un *Ambiente* che associa valori denotabili (fra cui le locazioni di memoria) a nomi ed una *Memoria* che associa locazioni di memoria a valori memorizzabili
- ☒ Nessuna delle altre risposte
- ☐ Non esistono valori denotabili
- ☐ Il comando di assegnamento non ha effetti collaterali

Domanda 2 β -riducendo $(\lambda a.((a\lambda b.\lambda c.c)\lambda d.\lambda e.d))(\lambda f.\lambda g.f)$ si ottiene:

- ☒ $\lambda b.\lambda c.c$
- ☐ Nessuna delle altre risposte
- ☐ La riduzione non termina
- ☐ c
- ☐ $\lambda b.\lambda c.b$

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

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



```
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 2: Esempio di pseudocodice

Domanda 4 Dato il frammento di programma (espresso in pseudo-codice) della Figura 2, quanto vale la variabile globale `c` dopo aver eseguito `topolino()`, assumendo scope dinamico?

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

Domanda 5 I puntatori di catena dinamica contenuti in un record di attivazione:

- ☐ Nessuna delle altre risposte
- ☐ Non esistono “puntatori di catena dinamica” in un record di attivazione
- ☐ Servono per identificare la zona di memoria in cui è memorizzata una variabile locale
- ☒ Devono essere esplicitamente allocati e deallocati dal codice del programma che li usa
- ☐ Servono per accedere alle variabili dinamiche



Domanda 6 β -riducendo $(\lambda x.xy)(\lambda z.zx)(\lambda z.zx)$ si ottiene:

- ☐ Nessuna delle altre risposte
- ☐ xyz
- ☐ $(\lambda x.xy)yx$
- ☒ $yx(\lambda z.zx)$
- ☐ La riduzione non termina

$(\lambda x.xy)(\lambda z.zx)(\lambda z.zx)$

Domanda 7 Un *compilatore* da un linguaggio \mathcal{L} ad un linguaggio \mathcal{L}_O è:

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

Domanda 8 I dangling pointer:

- ☐ Sono identificabili tramite tecniche di reference counting (contatore dei riferimenti)
- ☐ Sono un problema solo per il linguaggio Java
- ☒ Sono identificabili tramite la tecnica detta "mark and sweep"
- ☐ Non possono essere identificati con certezza, ma questo non è un problema perché comportano solo un piccolo spreco di memoria
- ☐ Nessuna delle altre risposte

Domanda 9 Dato il frammento di programma (espresso in pseudo-codice) contenuto in Figura 3, qual'è il valore di ritorno di $f1()$, assumendo scope statico?

- ☐ 5
- ☐ Nessuna delle altre risposte
- ☐ Non è possibile dirlo
- ☒ 0
- ☐ -5

Domanda 10 La *frammentazione esterna* causa:

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



```
int x, y, z;
void f3(void)
{
    x = 0;
    y = 5;
}

void f2(void)
{
    int y;

    f3();
    y = 0;
    z = 10;
}

int f1(void)
{
    int x;

    x = -5;
    y = 10;
    z = x + y;
    f2();

    return z - y - x;
}
```

Figure 3: Esempio di pseudocodice

Domanda 11 La *valutazione con corto circuito* del predicato “A && B” (dove “&&” rappresenta un and logico):

- ☐ Stabilisce che se “B” è vero allora “A” non viene valutato
- ☐ Nessuna delle altre risposte
- ☒ Stabilisce che se “A” è falso allora “B” non viene valutato
- ☐ Crea un non-determinismo nell’ordine della valutazione di “A” e “B”
- ☐ Stabilisce che “A” e “B” devono essere valutati in parallelo

Domanda 12 Se l’ambiente di una funzione non contiene il nome della funzione stessa:

- ☐ Nessuna delle altre risposte
- ☒ Non è possibile per la funzione invocarsi ricorsivamente
- ☐ Non ci sono particolari conseguenze
- ☐ La funzione non può usare scope dinamico
- ☐ La funzione non può usare scope statico