



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 Un'entità esprimibile è:

- ☐ Nessuna delle altre risposte
- ☐ Un'entità che può essere generata come risultato da un'espressione complessa o da una funzione
- ☐ Un'entità che ancora non compare nell'ambiente
- ☐ Un'entità che può essere memorizzata
- ☐ Una generica entità a cui può essere dato un nome

Domanda 2 In assenza di ambiente non locale:

- ☐ Per implementare funzioni definite ricorsivamente è necessario utilizzare un fixed point combinator
- ☐ Non si possono implementare algoritmi ricorsivi
- ☐ Nessuna delle altre risposte
- ☐ Non si possono implementare algoritmi iterativi o ricorsivi
- ☐ Non si possono implementare algoritmi iterativi

Domanda 3 Dato il frammento di programma (espresso in pseudo-codice) contenuto in Figura 1, qual'è il valore di ritorno di `f1()`, assumendo scope statico?

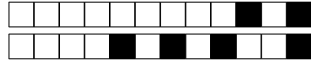
- ☐ Nessuna delle altre risposte
- ☐ Non è possibile dirlo
- ☐ 5
- ☐ -5
- ☐ 0

Domanda 4 β -riducendo $(\lambda a.((a\lambda b.\lambda c.c)\lambda d.\lambda e.d))(\lambda f.\lambda g.f)$ si ottiene:

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

Domanda 5 β -riducendo $(\lambda a.aaa)((\lambda b.b)(\lambda c.c))$ si ottiene:

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



```
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 1: Esempio di pseudocodice

Domanda 6 Si può dire che una macchina astratta che capisce il linguaggio C non sia implementata in modo puramente compilativo perché:

- ☐ Gli eseguibili generati da un compilatore C in genere non eseguono direttamente sulla macchina hardware, ma su una macchina astratta che include il *runtime* del linguaggio e le funzionalità del Sistema Operativo
- ☐ Il *runtime* del linguaggio C è comunque sempre interpretato
- ☐ Una macchina astratta che capisca un linguaggio di alto livello come il C non è mai implementabile con un compilatore
- ☐ Nessuna delle altre risposte
- ☐ Gli eseguibili generati dal compilatore vengono comunque interpretati da una macchina virtuale

Domanda 7 In presenza di variabili modificabili:

- ☐ Nessuna delle altre risposte
- ☐ 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
- ☐ Non esistono valori denotabili
- ☐ La valutazione del comando di assegnamento restituisce sempre un valore
- ☐ Il comando di assegnamento non ha effetti collaterali



```
int c = 2;

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

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

Figure 2: Esempio di pseudocodice

```
int mistero(int a, int b)
{
    if (b == 0) return a;
    return mistero(a / 2, b - 1);
}
```

Figure 3: Esempio di pseudocodice

Domanda 8 Si consideri lo pseudo-codice di Figura 2. Qual'è il valore di ritorno di `pluto()` se i parametri sono passati *per valore*?

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

Domanda 9 L'allocazione dinamica della memoria:

- ☐ E' sempre effettuata solo dal compilatore o dall'interprete
- ☐ Può essere fatta solo dallo *heap*
- ☐ Può essere fatta solo dallo *stack*
- ☐ Nessuna delle altre risposte
- ☐ Può essere fatta sia dallo *stack* che dallo *heap*

Domanda 10 La funzione implementata dallo pseudo-codice di Figura 3:

- ☐ Può causare una crescita incontrollata dello stack
- ☐ Non usa ricorsione in coda
- ☐ Nessuna delle altre risposte
- ☐ Causa sempre ricorsione infinita
- ☐ Non può essere implementata per via iterativa



Domanda 11 La *frammentazione interna* causa:

- ☐ 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
- ☐ Nessuna delle altre risposte
- ☐ Uno spreco di memoria
- ☐ Il funzionamento non corretto di programmi che allocano memoria dinamicamente

Domanda 12 Se gli array sono memorizzati *per colonne* ed `int a[100][100]` è un array multidimensionale di interi (si assuma che la dimensione di un intero sia 4 byte) con `a[0][0]` che ha indirizzo `0x5000`, qual'è l'indirizzo di `a[5][10]`?:

- ☐ `0x5510`
- ☐ `0x53ED`
- ☐ Nessuna delle altre risposte
- ☐ `0x500F`
- ☐ `0x41FE`