



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: ..... PRANZIIL MATTEO .....

Matricola: ..... 192198 .....

**Domanda 1** Se gli array sono memorizzati *per colonne* ed `short int a[100][100]` è un array multidimensionale di interi corti (si assuma che la dimensione di uno `short int` sia 2 byte) con `a[0][0]` che ha indirizzo `0x4100`, qual'è l'indirizzo di `a[5][10]`?:

- ☒ 0x48DA
- ☐ 0x4510
- ☐ Nessuna delle altre risposte
- ☐ 0x47DA
- ☐ 0x41FE

**Domanda 2** La *frammentazione interna* causa:

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

**Domanda 3** L'allocazione dinamica della memoria:

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

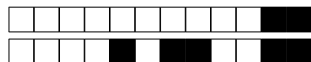
↑  
ESTERNA

**Domanda 4** Se l'ambiente di una funzione non contiene il nome della funzione stessa:

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

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

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



**Domanda 6** L'*ambiente* (o *environment*) è:

- ☒ L'insieme delle associazioni (nome, entità denotabile) esistenti in uno specifico punto del programma ed in uno specifico momento durante l'esecuzione di un programma
- ☐ L'insieme dei valori che una variabile assume durante l'esecuzione di un programma
- ☐ Un insieme di associazioni (nome, valore) definite staticamente durante lo sviluppo di un programma
- ☐ Nessuna delle altre risposte
- ☐ Una lista di coppie (nome, tipo) che permette di accedere alle variabili di un programma

**Domanda 7**  $\beta$ -riducendo  $(\lambda a.aaa)((\lambda b.b)(\lambda c.c))$  si ottiene:

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

$(\lambda b.b)(\lambda c.c)$  per 3 volte  
 $(\lambda c.c)$  per 3 volte  
 $\lambda c.c$  ←

**Domanda 8** La *valutazione con corto circuito* del predicato "A && B" (dove "&&" rappresenta un and logico):

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

**Domanda 9** Si consideri lo pseudo-codice di Figura 1. Qual'è il valore di ritorno di `pluto()` se i parametri sono passati *per valore*?

- ☒ Nessuna delle altre risposte
- ☐ 1
- ☐ 3
- ☐ 0
- ☐ Dipende dal tipo di scope (statico o dinamico) utilizzato

**Domanda 10** Funzione implementata dallo pseudo-codice di Figura 2:

- ☐ Non usa ricorsione in coda
- ☐ Non può essere implementata per via iterativa
- ☐ Nessuna delle altre risposte
- ☐ Causa sempre ricorsione infinita
- ☒ Usa ricorsione in coda

NO



```
int b = 666;

int pippo(int x) =X INUTILE
{
    x = 666; b = 1;
    return x / 2; 333
}

int pluto(void)
{
    int a, c; null, null

    a = b / 333; Q=2
    c = pippo(a); C=333

    return a + c; 333
}
```

Figure 1: Esempio di pseudocodice

```
int somma(int a, int b)
{
    if (a == 0) return b;
    return somma(a - 1, b + 1);
}
```

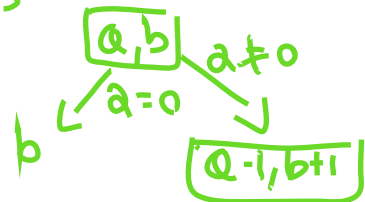


Figure 2: Esempio di pseudocodice

**Domanda 11** Dato il frammento di programma (espresso in pseudo-codice) della Figura 3, quanto vale la variabile globale `c` dopo aver eseguito `topolino()`, assumendo scope dinamico?

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

**Domanda 12** Si può dire che una macchina astratta che capisce il linguaggio Java non sia implementata in modo puramente compilativo perché:

- ☒ Non esistendo un vero e proprio *runtime* per Java, non si può parlare di compilazione pura JRE
- ☐ La macchina virtuale di Java (JVM) deve comunque essere compilata Bytcode interpret.
- ☐ Una macchina astratta che capisca un linguaggio di alto livello come Java non è mai implementabile con un compilatore NO
- ☐ Non esistono compilatori Java NO
- ☐ Nessuna delle altre risposte n.



```
int a, b, c;

void pippo(void)
{
    int a; 1
    a = 6;
    b = 5;
}

void pluto(void)
{
    int c; null
    int b; 10
    pippo();
    c = 3;
    a = 4;
}

void topolino(void)
{
    int a; 1
    a = 1;
    b = 10;
    pluto();

    c = a + b;
}
```

Figure 3: Esempio di pseudocodice