

Informatica Generale II - Prova teorica

A.A. 20052006

Esame: 25 gennaio 2007

Codice: OWTM

1. Il seguente frammento di codice:

```
{  
  double i=1.0;  
  double *j;  
  j=&i;  
  *j=10.0;  
  ...  
}
```

- (a) è errato perché si tenta di assegnare ad un puntatore un valore intero
- (b) è corretto
- (c) è errato perché l'operatore di dereferenziazione non può comparire a sinistra dell'operatore d'assegnamento
- (d) è errato perché j punta a una zona di memoria arbitraria
- (e) non rispondo

2. Quale è la lunghezza del cammino dell'albero in figura 1

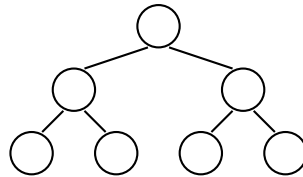


Figura 1:

- (a) 7
 - (b) 2
 - (c) 11
 - (d) 10
 - (e) non rispondo
3. Le operazioni di base di stack e code sono:
- (a) push e pop per stack, put e get per code
 - (b) push e pop per code, put e get per stack
 - (c) put e get
 - (d) push e pop
 - (e) non rispondo
4. Quale delle seguenti affermazioni riguardanti l'ordinamento per selezione è *falsa*?
- (a) è composto da due cicli `for` che vengono eseguiti un numero di volte che dipende dai particolari dati d'ingresso;

- (b) l'algoritmo impiega $\mathcal{O}(n^2)$ operazioni anche se l'array è già completamente ordinato;
 - (c) ha complessità asintotica $\mathcal{O}(n^2)$ nel caso peggiore e in quello migliore;
 - (d) è composto da due cicli `for` che vengono eseguiti un numero di volte che non dipende dai particolari dati d'ingresso;
 - (e) non rispondo
5. Si consideri la rappresentazione di un grafo mediante *matrice delle adiacenze* (sia n il numero di nodi). Quale tra le seguenti affermazioni è *falsa*?
- (a) l'elemento nella riga i e nella colonna j della matrice è pari a 1 se nel grafo rappresentato c'è un arco dal nodo i al nodo j , è pari a 0 nel caso contrario;
 - (b) tale rappresentazione richiede un'occupazione di memoria proporzionale al numero massimo di archi del grafo;
 - (c) l'operazione di accesso ai successori di un nodo richiede l'accesso ad n elementi della matrice;
 - (d) tale rappresentazione richiede un'occupazione di memoria proporzionale al numero di nodi del grafo;
 - (e) non rispondo
6. Dato il grafo in figura 2:

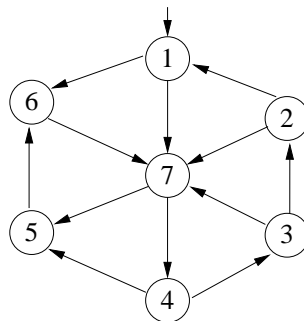


Figura 2:

- (a) La visita in ampiezza produce la seguente sequenza d'uscita: 1 6 7 5 3 4 2
 - (b) La visita in ampiezza produce la seguente sequenza d'uscita: 1 6 7 4 5 3 2
 - (c) Nessuna delle risposte è accettabile;
 - (d) La visita in ampiezza produce la seguente sequenza d'uscita: 1 7 4 3 2 5 6
 - (e) non rispondo
7. Si analizzi il seguente codice:

```

void foo(int A[], int N) {
    for (int i=0; i < N-1; ++i) {
        int min = i;
        for (int j=i+1; j < N; ++j)
            if (A[j] < A[min]) min = j;
        swap(A[i], A[min]);
    }
}

```

Esso implementa l'algoritmo:

- (a) Mergesort
- (b) Quicksort
- (c) Bubblesort
- (d) Selection sort
- (e) non rispondo

8. Supponendo di aver definito una funzione funzEsame con la seguente intestazione:

```
void funzEsame (double *i, double *j);
```

L'invocazione della suddetta funzione funzEsame avrà la forma:

- (a) `double i=1.0, j=0.5;`
`funzEsame(i, j);`
- (b) `double i=1.0, j=0.5;`
`funzEsame();`
- (c) `double i=1.0, j=0.5;`
`funzEsame(*i, *j);`
- (d) `double i=1.0, j=0.5;`
`funzEsame(&i, &j);`
- (e) non rispondo

9. Dato l'albero n-ario in figura 3:

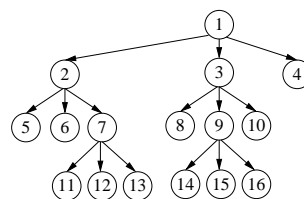


Figura 3:

- (a) La visita in postordine dell'albero produce in uscita la seguente sequenza: 5 6 11 12 13 7 2 8 14 15 16 9 10 3 4 1
- (b) Non si può effettuare la visita in postordine di un albero non binario;
- (c) La visita in postordine dell'albero produce in uscita la seguente sequenza: 5 6 7 11 12 13 2 8 14 15 16 9 10 3 4 1
- (d) La visita in postordine dell'albero produce in uscita la seguente sequenza: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
- (e) non rispondo

10. Il seguente frammento di codice:

```
struct data {
    int giorno;
    int mese;
    int anno;
    data(int g, int m, int a){ giorno=g; mese=m; anno=a; }
};

struct persona {
    char* nome;
    char* cognome;
    data datanascita;
    persona(char * n, char * c, data d_nasc) {
        nome=n; cognome=c; datanascita=d_nasc;
    }
};

int main() {
    ...
    char n[]="Sergio";
```

```

char c[]="Rossi";
data datanasc(25,12,1988);
persona io(n, c, datanasc);
...
}

```

- (a) produce un errore a tempo d'esecuzione perché l'istruzione `persona io(n, c, datanasc);` richiama il costruttore senza argomenti di `data` che non è presente all'interno della definizione di `data`;
- (b) è corretto;
- (c) produce un errore a tempo di compilazione. Infatti, all'interno della dichiarazione di `persona` si ha l'invocazione del costruttore standard senza argomenti di `data` che però è stato inibito automaticamente dal compilatore a fronte della dichiarazione del costruttore a tre argomenti;
- (d) produce un errore a tempo d'esecuzione. Infatti, all'interno della dichiarazione di `persona` si ha l'invocazione del costruttore standard senza argomenti di `data` che però è stato inibito automaticamente dal compilatore a fronte della dichiarazione del costruttore a tre argomenti;
- (e) non rispondo