

Prova Scritta di Programmazione e Laboratorio - 15 Settembre 2021

Valutazione: 2 punti per ogni risposta esatta; -0.5 punti per ogni risposta errata; 0 punti per non risposta;
Massimo 6 punti per gli esercizi di programmazione;

Per ogni domanda a risposta multipla, marcare la risposta esatta su questo foglio;

E' possibile usare un foglio aggiuntivo per le risposte di programmazione.

Tempo a disposizione: 2 ore.

Cognome e Nome: _____ Matricola: _____

Modulo 1

1. Dire a quale dei codici che utilizzano il ciclo for corrisponde il seguente codice

```
int a,c=0;
scanf("%d",&a);
while(a!=0){
    if (a<0)
        c=c+1;
    scanf("%d",&a);}

```

- A. `for(c=0,scanf("%d",&a);a!=0;c++,scanf("%d",&a));`
- B. `for(c=0,a=1;a;c--=(a>>31),scanf("%d",&a));`
- C. `for(c=0,a=1;a!=0;c++,scanf("%d",&a));`
- D. `for(c=-1,a=1;a;c = a<0 ? c=c+1 : c,scanf("%d",&a));`

2. Sia il seguente sorgente il cui nome è `main.c`

```
#include <stdio.h>
int main(){
    float a=-0.625;
    unsigned int b = *((unsigned int *)&a);
    printf("%d\n", (b&0xff8000000)>>23);}

```

Si consideri la compilazione del sorgente nell'eseguibile `main`. L'output del comando `main` sarà:

- A. 126
- B. 382
- C. Il codice non è corretto, non è possibile eseguire la compilazione
- D. -1

3. Si considerino gli algoritmi di sorting svolti a lezione durante il primo modulo di programmazione. Dire quali tra queste affermazioni è falsa

- A. Non sono tutti di complessità quadratica sulla taglia dell'array;
- B. Non sono tutti basati su scambi tra elementi;
- C. Ve ne sono di non generali (non applicabili a qualsiasi tipo di dato);
- D. Quelli generali sono sempre basati su scambi tra elementi.

4. Il complemento a 2 della variabile `unsigned char a = 0x0a` è:

- A. 246_{10} ;

- B. 1010_2 ;
- C. $f5_{16}$;
- D. Nessuno dei precedenti

5. Sia la seguente funzione:

```
void fun(int *A, int n){
    int v,imin=0,imax=n-2;
    int b=A[n-1];
    while((imin<imax)&&(*(A+(v=imin+((imax-imin)>>1)))-b)){
        imin = *(A+v)>b ? imin : v+1;
        imax = *(A+v)<b ? imax : v-1;}}}
```

Dire per quali dei seguenti input la funzione `fun` esegue il numero minimo di confronti

- A. `int A[]={0,1,2,4,5,7,8,10,9,11,25}; myfun(A,9);`
 - B. `int A[]={0,2,4,6,8,10,12,15,18,16,-21}; myfun(A,10);`
 - C. `int A[]={2,2,2,3,4,5,6,8,8,10,11,12,7,31}; myfun(A,12);`
 - D. `int A[]={0,1,2,4,5,7,9,10,10,10,2,-1,-20}; myfun(A,11);`
6. Una matrice quadrata A di m elementi é detta *simmetrica* se $A(i,j) = A(j,i)$ per $i \neq j$. Data la matrice Z a tre dimensioni che memorizza n matrici bidimensionali $m \times m$ di valori `double`, si richiede di scrivere la funzione:

```
void simmetriche(double ***Z, int n, int m, int *ns)
```

che ritorna nell'argomento `*ns` il numero di matrici bidimensionali simmetriche memorizzate in Z .

Modulo 2

1. Il mergesort naturale bottom-up su liste concatenate
 - A. fonde iterativamente sottoliste in ordine crescente con sottoliste in ordine decrescente;
 - B. trova il puntatore all'elemento centrale della lista e procede ricorsivamente su due sottoliste;
 - C. usa un array di liste precaricato con le sottoliste ordinate della lista di partenza;
 - D. fonde iterativamente sottoliste di lunghezza uno, poi di lunghezza due, eccetera.
2. Per cancellare da un albero binario di ricerca un nodo **r** dotato di entrambi i figli sinistro e destro bisogna
 - A. trovare il predecessore **x** di **r**, cancellare il massimo nel sottoalbero destro di **r**, e mettere **x** al posto di **r**
 - B. trovare il successore **x** di **r**, cancellare il minimo nel sottoalbero sinistro di **r**, e mettere **x** al posto di **r**
 - C. trovare il predecessore **x** di **r**, cancellare il massimo nel sottoalbero destro di **r**, e mettere **x** al posto di **r**
 - D. trovare il successore **x** di **r**, cancellare il minimo nel sottoalbero destro di **r**, e mettere **x** al posto di **r**
3. L'interfaccia di funzione
`void ordinamento(void **vettore, int length)`
 - A. è appropriata per una funzione che ordina dati di tipo intero;
 - B. non può essere relativa ad una funzione che ordina un vettore di tipo qualsiasi, perché non ha tra gli argomenti un puntatore a funzione di comparazione;
 - C. può essere relativa ad una funzione che ordina un vettore di tipo qualsiasi, solo se priva di chiamate ad altre funzioni;
 - D. è appropriata per una funzione che ordina un vettore di tipo qualsiasi, eventualmente usando una funzione di comparazione di elementi definita esternamente al file.
4. La direttiva

```
#define UNSIGNED_BIT CHAR_BIT * sizeof( unsigned int )
```

- A. definisce `UNSIGNED_BIT` come il numero di bit di un intero senza segno;
 - B. definisce `UNSIGNED_BIT` come puntatore a carattere;
 - C. definisce `UNSIGNED_BIT` come numero di byte di un intero senza segno;
 - D. assegna a `UNSIGNED_BIT` un intero senza segno moltiplicato per il numero di bit di un carattere.
5. Quale delle seguenti funzioni cancella la più piccola chiave di un albero binario di ricerca?
 - A.

```
static void deleteMin(TreeNodePtr *treePtr) {  
    if ((*treePtr)->left == NULL) {  
        TreeNodePtr tmp = (*treePtr)->right;  
        free(*treePtr); *treePtr = tmp;  
    }  
    else deleteMin(&((*treePtr)->left));  
}
```

- B. `static void deleteMin(TreeNodePtr *treePtr) {`
 `if ((*treePtr)->right == NULL) {`
 `TreeNodePtr tmp = (*treePtr)->left;`
 `free(*treePtr); *treePtr = tmp;`
 `}`
 `else deleteMin(&((*treePtr)->left));`
 `}`
- C. `static void deleteMin(TreeNodePtr *treePtr) {`
 `if ((*treePtr)->left == NULL) {`
 `TreeNodePtr tmp = (*treePtr)->right;`
 `free(*treePtr); *treePtr = tmp;`
 `}`
 `else deleteMin(&((*treePtr)->right));`
 `}`
- D. `static void deleteMin(TreeNodePtr *treePtr) {`
 `if ((*treePtr)->right == NULL) {`
 `TreeNodePtr tmp = (*treePtr)->left;`
 `free(*treePtr); *treePtr = tmp;`
 `}`
 `else deleteMin(&((*treePtr)->right));`
 `}`

6. Costruire una funzione

```
void valuta(struct node *l)
```

che, ricevuta in ingresso una lista di nodi di tipo

```
struct node {
    int (*fun)(int a);
    int arg;
    struct node *nextPtr;
};
```

valuti la funzione `fun()` sull'argomento `arg` per tutti gli elementi della lista `l`. Si inserisca la funzione `valuta()` in un semplice programma di prova.