

NOME:

COGNOME:

MATRICOLA:

**[1] Punti 2 – Domanda a risposte multiple (sono possibili 1 o più risposte)**

Dati le seguenti dichiarazioni di file:

```
//file1.cc
struct ss{int a;} ;
void fun(ss b) { ... }

//file2.cc
struct ss {char c ; int z; };
void fun(ss);
int main() {ss k; fun(k); }
```

- a. Il compilatore segnala errore perché rileva due dichiarazioni incompatibili della struttura `ss`
- b. Il compilatore segnala errori perché in `file2.cc` non è stata usata la parola chiave `extern` per riferirsi alla dichiarazione di `ss`
- c. Il compilatore segnala errore perché l'identificatore `ss` non ha collegamento esterno
- d. Nessuna delle precedenti

**[2] Punti 3 – Domanda a risposta aperta**

Cosa stampa il seguente programma: \_\_\_\_\_

```
void f(int a, int &b, int *&c, int *d) {
    a*=2;
    b+=a;
    c=d;
    *c = 1;}

int main() {
    int a = 7, b = 15;
    int *c = &b;
    int *d = &a;
    f(a, b, c, d);
    cout <<a<< " " <<b<< " " <<*c<< " " <<*d<< " " << endl;}
```

**[3] Punti 3 – Domanda a risposta aperta**

Cosa stampa il seguente programma: \_\_\_\_\_

```
#include <iostream>

using namespace std ;

int f(int);

int main(){
    cout<<"f(4)="<<f(4)<<endl;
    return 0;
}
```

NOME:

COGNOME:

MATRICOLA:

```
int f(int n){
    if ((n-1)<=0)
        return n;
    return f(n-1)+f(n-2);
}
```

**[4] Punti 3 – Domanda a risposta aperta**

Cosa stampa il seguente programma: \_\_\_\_\_

```
#include <iostream>
using namespace std ;

int main()  {

    int v[4]={1,2,3,4};
    int *p = v+1;
    int i;

    *(p+2)=6;
    *p=(*p)+2;
    for(i=0;i<4;i++)
        cout<<v[i]<< " ";
}
```

**[5] Punti 4 - Scrittura di codice**

Data una sequenza di valori interi memorizzati in una lista doppia di elementi di tipo `elem`, si scriva la *funzione* `bool max(elem* e)` che, dato un elemento `e` della lista (la posizione nella lista non è nota), restituisca `true` se il valore registrato in `e` è il valore massimo, `false` altrimenti.

```
struct elem
{
    int inf;
    elem* pun;
    elem* prev;
};

bool max(elem* e){
```

```
}
```

NOME:

COGNOME:

MATRICOLA:

**[6] Punti 8 - Scrittura di codice**

Data la seguente dichiarazione di lista e primitive

```
struct elem
{
    int inf ;
    elem* pun ;
} ;

typedef elem* lista ;

int head(lista); //restituisce il contenuto della testa
lista tail(lista); //restituisce la coda della lista
```

**a. Punti 4**

Si scriva la *procedura ricorsiva* `void stampa_inv(lista)` che, dato in input la testa della lista, stampa l'elenco degli elementi in ordine inverso

```
void stampa_inv(lista l){
```

```
}
```

**b. Punti 4**

Si scriva la funzione `lista sposta(lista& l, int soglia)` che sposta dalla lista l tutti gli elementi i cui valori sono sotto il valore soglia e restituisce la lista degli elementi spostati. La funzione *non deve fare allocazioni*. Ad esempio data la lista [1,2,3,4] e il valore di soglia 3 la funzione restituisce la testa della lista [1,2] mentre l diventa [3,4].

```
lista sposta(lista& l, int soglia){
```

NOME:

COGNOME:

MATRICOLA:

}

**[7] Punti 9 - Scrittura di codice**

Dato la segue dichiarazione

```
struct libro{
    char* autore;
    char* titolo;
};
```

**a. Punti 3**

Si scriva la funzione `int compare(libro, libro)` che implementa la seguente relazione d'ordine:

`compare(l1,l2)=0` se autore e titolo coincidono

`compare(l1,l2)<0` se l'autore di l1 precede l'autore di l2 e oppure l'autore è lo stesso e il titolo di l1 precede il titolo di l2

`compare(e1,e2)>0` altrimenti

```
int compare(libro l1, libro l2){
```

}

**b. Punti 2 – Scrittura di codice**

Si assuma un BST con chiave di tipo `libro`. Scrivere il tipo di dato `bnode` (nodo del BST) e il tipo di dato `bst` (puntatore alla radice del BST)

NOME:

COGNOME:

MATRICOLA:

**c. Punti 4**

Scrivere la *procedura* `void aggiorna_libro(bst& b, libro* l, char* a)` che aggiorna il contenuto del libro `l` sostituendo l'autore con la stringa contenuta in `a`. La funzione deve usare la funzione `compare` (punto a).

```
void aggiorna_libro(bst& b, libro* l, char* a) {
```

```
}
```