



UNIVERSITÀ DEGLI STUDI DI PISA

FACOLTÀ DI INGEGNERIA

CORSO di LAUREA in INGEGNERIA delle TELECOMUNICAZIONI

ESAME DI FONDAMENTI DI INFORMATICA - SIMONCINI – 11/01/2008 - C++

1. Si definisca una classe `Vett`, che consenta le seguenti operazioni su un vettore `A` a valori interi:
 - `Vett(n)`: costruttore che crea il vettore `A` di dimensione uguale a `n` inizialmente a valori nulli.
 - `int massimo()`: restituisce l'indice di `A` corrispondente al valore massimo in esso contenuto (se esistono due o più elementi di `A` con valore massimo, si restituisce l'indice di valore maggiore).
 - `Modifica(i, x)`: modifica l'elemento di `A` di indice `i` aggiungendogli il valore intero `x`.
 - `int positivi()`: restituisce il numero di elementi di `A` con valore maggiore di 0.
 - `~Vett()`: distruttore.
2. Sia data una lista semplice `l` a valori interi diversi fra loro. Si scriva una funzione `modifica(l, x)` che riceve in ingresso la lista `l` e un valore intero `x`. Se `l` contiene un elemento con campo informativo uguale a `x`, la funzione lo elimina. Altrimenti la funzione inserisce in coda un nuovo elemento con campo informativo uguale a `x`.
3. Sia dato un file `numeri.txt` che contiene un elenco non vuoto di valori interi. Scrivere una funzione `crea_intorno(j)` che riceve in ingresso un valore intero `j`. La funzione calcola la media aritmetica `AV` dei valori di `numeri.txt` e crea un nuovo file `intorno.txt` composto dai valori di `numeri.txt` compresi nell'intervallo `[AV-j, AV+j]`.
4. Dati i numeri decimali `A=129`, `B=97` e `C=-110`, si determini il minimo numero di bit necessari per rappresentare contemporaneamente i suddetti numeri in complemento a 2 e se ne dia la rappresentazione. Utilizzando poi lo stesso numero di bit, si esegua l'operazione `D=B-C` e si discuta se il risultato ottenuto è o no significativo.

Punteggio:

Es.1	Es.2	Es.3	Es.4
10	10	10	3



UNIVERSITÀ DEGLI STUDI DI PISA
FACOLTÀ DI INGEGNERIA
CORSO di LAUREA in INGEGNERIA ELETTRONICA

Es. 1

// VETT.h

```
class Vett {
    int n;
    int *A;
public:
    Vett(int n);
    int Massimo();
    void Modifica(int l, int x);
    int positivi();
    ~Vett();
}
```

//VETT.cc

#include "VETT.h"

```
Vett::Vett(int n) {
    A=new int[n];
    for(int l=0; l<n; l++) A[l]=0;
}
```

```
int Vett::Massimo() {
    int max=A[0];
    int i_m;
    for(int i=1; i<n; i++)
        if(A[i]>=max) { max=A[i];
                       i_m=i; }

    return i_m;
}
```

```
void Vett::Modifica(int l, int x) {
    if( l<n && l>=0) A[l]+=x;
}
```

```
int Vett::positivi() {
    int cont=0;
    for(int i=0 ; i<n ; i++)
        if(A[i]>0) cont++;

    return cont;
}
```

```
Vett::~~Vett() { delete A[]; }
```

Es. 2

```
struct elem {
    int info;
    elem *next};
```

```
void modifica(elem *&l, int x)
{
    elem * aux1, *aux2;
```

```
    if(l==0) {
        aux1=new elem;
        aux1->info=x;
        aux1->next=0;
        l=aux1 ;
```



UNIVERSITÀ DEGLI STUDI DI PISA
FACOLTÀ DI INGEGNERIA
CORSO di LAUREA in INGEGNERIA ELETTRONICA

```
        return;
    }

    aux1=l;
    if(aux1->info==x) {
        l=aux1->next ;
        delete aux1 ;
        return ;}
    for(aux2=aux1->next ; aux2 !=0 && aux2->info !=x; aux2=aux2->next)
        aux1=aux2 ;

    if(aux2==0) { //inserimento in coda
        aux2=new elem
        aux2->info=x ;
        aux1->next=aux2 ;
    }
    else { //eliminazione
        aux1->next=aux2->next ;
        delete aux2;
    }
}
```

Es. 3

```
#include <fstream.h>

void crea_intorno (int j) {
    int n, cont=0, somma=0;
    int AV;

    fstream f, g;
    f.open("numeri.txt", ios::in);
    while(f>>n) {
        cont++;
        somma+=n;
    }

    AV=somma/cont;

    f.close();
    f.open("numeri.txt", ios::in);
    g.open("intorno.txt", ios::out);

    while(f>>n)
        if(n<=AV+j && n>=AV-j)    g<<n<<"\n";

    f.close();
    g.close();
}
```