



UNIVERSITÀ DEGLI STUDI DI PISA  
FACOLTÀ DI INGEGNERIA  
CORSO di LAUREA in INGEGNERIA delle TELECOMUNICAZIONI

**ESAME DI FONDAMENTI DI INFORMATICA 29/01/2009 - C++**

- 1. Scrivere una classe `Vett` che gestisce il contenuto di un vettore `V` a valori interi di dimensione variabile `n` (`n=10` per default). Le possibili operazioni sul vettore `V` sono:**
  - `vett(int n)`: costruttore che crea il vettore `V` di dimensione `n` a valori nulli.
  - `vett()`: costruttore di default che crea `V` di dimensione 10 a valori nulli.
  - `bool controllo()`: controlla se tutti gli elementi di `V` hanno un valore positivo.
  - `modifica(int i, int x)`: modifica l'elemento `i`-esimo di `V` sommandogli `x`.
  - `int max()`: restituisce il valore massimo contenuto in `V`.
  - `~vett()`: distruttore.
- 2. Dato un file di testo `paper.dat` composto da almeno 10 righe non vuote. Si scriva una funzione `finali()` che costruisce e stampa sul monitor una stringa `s` composta dall'ultimo carattere di ciascuna delle prime 10 righe di `paper.dat`.**
- 3. Data una lista semplice `l` a valori interi non vuota, si scriva una funzione `rendi_pari(l)` che riceve in ingresso la lista `l` e conta gli elementi di `l`, se questi sono in numero dispari, allora elimina l'elemento centrale della lista.**
- 4. Dati i numeri decimali `A=127`, `B=101` e `C=-227`, 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

1.

```
//vett.h
class vett {
    int *V;
    int dim;
public:
    vett(int n);
    vett ();
    bool controllo();
    void modifica(int i; int x);
    int max();
    ~vett();}

//vett.cc
#include "vett.h"

vett::vett(int n) {
    V=new int[n];
    dim=n;
    for(int i=0; i<dim; i++)
        V[i]=0;
}

vett::vett() {
    V = new int[10];
    dim=10;
    for(int i=0; i<10; i++)
        V[i]=0;
}

bool vett::controllo() {
    for(inti=0; i<dim; i++)
        if(V[i]<=0) return false;

    return true; }

void vett::modifica(int i, int x) {
    V[i]+=x; }

int vett::max() {
    int M;
    M=V[0];
    for(int i=1; i<dim; i++)
        if(V[i]>M) M=V[i];
    return M; }

vett::~vett() {
    delete V[]; }
```

2.

```
#include <fstream.h>

void finali() {
    fstream f;
    f.open("paper.dat", ios::in);
```



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

```
char c1, c2;
char s[11];
int i=0;
f.get(c1);
f.get(c2);
while(!f.eof()) {
    if(c2!='\n') {    c1=c2;
                    f.get(c2);}
    else {    s[i]=c1;
            c1=c2;
            f.get(c2);
            i++;}
    if(i==10) break;
}
s[i+1]='\0';
cout<<s;
}
```

3.

```
void rendi_pari(elem *& l)
{
    elem *aux1, *aux2;
    aux1=l;
    int cont=0;
    while(aux1!=0) {
        cont++;
        aux1=aux1->next;
    }
    if(cont==1) {    delete l ;
                  l=0 ;
                  return}

    if(cont%2==0) return;

    aux1=l;
    aux2=l->next;
    for(int i=1; i<=cont/2; i++)
        {    aux1=aux2;
          aux2=aux2->next;
          }

    aux1->next=aux2->next;
    delete aux2;
}
```