VI Appello 23-07-2014 Soluzione

Esercizio 1

```
void inserisci(elem*& p0, char c) {
// inserimento in ordine crescente senza duplicati

elem*p, q;
for(q=p0; q!= NULL && q->info < c; q=q->pun)
  p=q;
if (q!= NULL && q->info == c)
  return;

elem*r = new elem;
r->info = c;
r->pun = q;
if (q==p0)
  p0=r;
else
  p->pun = r;
}


elem* nuovaLista(const char* nome){
 ifstream in(nome);
 elem* L = NULL;
 char carattere;
 while (in>>carattere)
   inserisci(L, carattere);
 return L;
}
```

Esercizio 2

```
punto* nuovoVettore(double* v1, int d1, double* v2, int d2){

int dim = d1;
if (d2>d1) dim=d2;

punto* v = new punto [dim];

for(int i=0; i<dim; i++){
// inizializzo campo x della
// struttura punto dell'elemento i-esimo
 if (i<d1)
    v[i].x = v1[i];
   else
    v[i].x = 0;

// inizializzo campo y della
// struttura punto dell'elemento i-esimo
if (i<d2)
    v[i].y = v2[i];
   else
    v[i].y = 0;
}
return v;
}
```

Esercizio 3

```
int contamultipli(elem* L) {
 if (L ==0) retun 0;
 if (((L->info)%13)==0)
     return 1 + contamultipli(L->pun);
 return  contamultipli(L->pun);

}
```

Esercizio 4

4.1) 2210 in base 3 corrisponde a 54+18+3=75 in base 10.

75 in base 10 corrisponde a 83 in base 9.

Un altro modo di fare questa conversione, senza passare
per la base 10, consiste nel raggruppare le cifre in base
3 in gruppi di due e poi utilizzare la seguente tabella
di conversione, usando un metodo simila a quando si converte
da base 2 a base 4:

22 => 8
21 => 7
20 => 6
12 => 5
11 => 4
10 => 3
02 => 2
01 => 1
00 => 0

Infatti:
22-10 => 8-3, cvd.

4.2)

Soluzione del programmino assembler

Siano X la terzultima cifra del proprio numero di matricola
e Y l'ultima.
Dopo aver letto XY in AL (ad esempio, se X=2 e Y=9 in AL avremo 0010-1001),
si effettua l'AND con FC, ossia con 1111-0110,
che ha come effetto quello di portare a zero il quartultimo e l'ultimo
bit di AL.
Il successivo OR con 0C, ossia con 0000-0110, fa si che il
terzo ed il secondo bit di AL diventino certamente 1 se non lo erano già.
Dunque la prima chiamata alla outbyte stampa a video "XC", con X
sempre la terzultima cifra.
A questo punto viene chiamata la soubroutine, che aggiunge 3 ad AL.
Pertanto il risultato e' XF. Successivamente AL viene ruotato a sinistra
di quattro. Pertanto l'ultima chiamata a outbyte stamperà "FX",
ossia F0, F1, F2, ecc... a seconda della propria terzultima cifra.

Riassumendo, quello che viene mostrato a video è (indipendentemente da Y):
X=0Y => "0CF0"
X=1Y => "1CF1"
X=2Y => "2CF2"
ecc...