Esercizio 1

```
bool lettera(char c) {
if ((c>= 'A') && (c <= 'Z')) || ((c>= 'a') && (c <= 'z')))
    return true ;
return false ;
}

bool cifra(char c) {
if ((c>= '0') && (c <= '9'))
    return true ;
return false ;
}

elem* creaLista(elem* testa) {
if (testa == NULL)
    return NULL;
elem* nuova = new elem ;
nuova->info = testa->info ;
nuova ->pun = NULL ;

elem* prec, *r, * q, *n ;

for(q=testa->pun; q !=NULL ; q=q->pun) {
    n=new elem ;
    n->info = q->info;
    n->pun = NULL ;

    if (lettera(n->info)) /* inserimento in testa */
        { n->pun = nuova ;
          nuova = n ;
        }
    else if (cifra(n->info)) /* inserimento dopo le lettere */
        { for(r= nuova ; (r != NULL) && lettera(r->info) ; r=r->pun)
            prec = r;
          if (r==nuova) {
              n->pun = nuova ;
              nuova = n ;
          }
          else {
              prec->pun = n ;
              n->pun = r ;
          }
        }
    else /* inserimento in fondo */
        { for(r= nuova ; (r->pun != NULL); r=r->pun);
          r->pun = n ;
        }
    return nuova;
}
```

Esercizio 2

```
char** leggi_stringhe(int N, int K) {
char** v;
if (N<=0) N = 5;
if (K<=0) K = 99;

v = new char*[N];
for(int j=0;j<N;j++) {
    v[j]= new char[K+1];
    cin >> v[j];
    if (!cin)
        strncpy(v[j], "Ciao", K);
}

return v;
}
```

Esercizio 3

```
void stampa(int n, int i=0) {

if ((n<=0) || ((n%2)==0)) {
    return;
}

for(int h=1; h<=i; h++)
    cout << ' ' ;

cout << '*';
if(n==1)
    return;
for(int k=1; k<= n-2; k++)
    cout << ' ' ;
cout << '* ' << endl;

stampa(n-2, i+1);
}
```

Esercizio 4.1

- a) Si: 0x34 e' il codice ASCII associato alla cifra '4'
- b) 11001011 in complemento a due rappresenta il numero negativo
meno $2^8 - (128+64+8+2+1) = \text{meno } 256-203 = \text{meno } 53 \Rightarrow -53$
- c) Si: il valore assunto corrisponde al decimale $32+16+4 = 52$

Esercizio 4.2 (assembler)

Il programma, dopo aver letto da tastiera la terz'ultima e la penultima cifra, ne mette la codifica esadecimale su 4 bit sulla parte alta e bassa del registro AL, rispettivamente.

L'AND logico fra il naturale 15 (ossia l'esadecimale 0x0F) ha l'effetto di azzerare la parte alta di AL.

Dopo aver incrementato il registro AL si avrà che

AH conterrà 0 e AL la codifica binaria dei numeri fra 1 e 10, a seconda di quella che era la penultima cifra della matricola dello studente: 0=>1, 1=2,..., 9=>10.

La successiva chiamata alla outbyte visualizzerà sullo schermo pertanto la seguente coppia di caratteri:

"01", "02", ..., "09", "0A", a seconda che la penultima cifra della matricola fosse, rispettivamente, 0,1,...,8, o 9.

A questo punto il programma passa a stampare una sequenza di asterischi lunga da 1 a 10, sempre in dipendenza del penultimo numero di matricola.

In conclusione, se X e' la terzultima cifra del proprio numero di matricola, si avra' una delle seguenti uscite (in relazione alla propria penultima cifra):

```
X0 => "01*"
X1 => "02**"
X2 => "03***"
X3 => "04****"
X4 => "05*****"
X5 => "06*****"
X6 => "07*****"
X7 => "08*****"
X8 => "09*****"
X9 => "0A*****"
```