

Esercizio 1

```

void inserisci(elem* & testa, const char* s) {
ifstream in;
in.open(s);
char c;
while(in >>c) {
    if( (c>= 'a' && c <= 'z') || (c>= 'A' && c <= 'Z')) {
        elem* r = new elem;
        r->info = c;
        elem *p, *q;
        for (q=testa; (q!= NULL) && (q->info < c); q=q->pun)
            p=q;
        if (q==testa) {
            r->pun = testa;
            testa = r;
        }
        else {
            p->pun = r;
            r->pun = q;
        }
    } //fine if
} // fine while
in.close();
}

```

Esercizio 2

```

int* nuovamatrice(int* m, int r, int c, int k) {
    int* m1 = new int [r*c] ;
    for(int i=0; i< r; i++)
        for (int j=0; j<c; j++) {
            int A=0, B=0;
            for (int s=1; s<=k; s++) {
                if (j+s < c)
                    A+= m[i*c + j + s] ;
                if (j-s >= 0)
                    A+= m[i*c + j - s] ;
            } // fine somma riga

            for (int t=1; t<=k; t++) {
                if (i+t < r)
                    B+= m[ (i+t) *c + j] ;
                if (i-t >= 0)
                    B+= m[(i-t)*c + j] ;
            } // fine somma colonna
            m1[i*c + j] = A*B ;
        }
    return m1;
}

```

Esercizio 3

stampaX(5,0);

```
void stampaX(int n, int k){
    if (k==n)
        return;
    for (int j=0; j<n; j++){
        if ((j==k) || (j==n-k-1) )
            cout<<"*";
        else
            cout<<" ";
    }
    cout<<endl;
    stampaX(n, k+1);
}
```

Esercizio 4

Prima iterazione

Il programma aggiunge 2 a FB, ottenendo FD e non dando overflow.

Dunque non salta a lab2, bensì continua con lab1 dove stampa il carattere 'D'.

Seconda iterazione

Il programma aggiunge 2 a FD, che produce FF.

Non si salta dunque a lab2, e pertanto si stampa ancora 'D'

Terza iterazione

Il programma aggiunge 2 a FF, generando un overflow.

Si salta dunque su lab2 dove viene stampato il codice ascii corrispondente al naturale 84, ossia il carattere 'T'.

In definitiva l'uscita prodotta a video sarà la seguente: DDT

Esercizio 4

Input -1

-1

0

ecc. char da g

fine g

10

fine main

Input 0

0

1

int da main

fine main

Input 1

1

ecc. char da main

fine main