Esercizio 1 (svolto)

- Scrivere una funzione che, dato un carattere passato in ingresso (come parametro formale), restituisca il carattere stesso se non è una lettera minuscola, altrimenti restituisca il corrispondente carattere maiuscolo

- Prima di definire il corpo della funzione, scrivere un programma che, usando SOLO tale funzione, legga un carattere da *stdin* e, se minuscolo, lo ristampi in maiuscolo, altrimenti comunichi che il carattere non è minuscolo
  - Adottiamo cioè, per esercizio, un approccio *top-down*

Specifiche della funzione

- Per adottare in modo efficace l'approccio top-down bisogna definire in modo esatto cosa va in ingresso alla funzione e cosa la funzione restituisce
  - Scriviamo quindi solo la dichiarazione della funzione
  - Inseriamo i dettagli sul comportamento della funzione sotto forma di commenti all'intestazione della funzione stessa

Prima parte

```
/*
 * Dato il carattere in ingresso c restituisce il maiuscolo
 * di c utilizzando solo le proprietà di ordinamento dei
 * codici dei caratteri.
 * Assunzione: se c non è minuscolo, ritorna il
 * carattere inalterato.
 */
char maiuscolo(char c);

main() {
  char minus, maius;
  cin>>minus;
  maius = maiuscolo (minus);
  if (minus==maius)
       cout<<"Il carattere "<<minus
           <<" non è minuscolo"<<endl;
  else
       cout<<"Minuscolo = "<<minus<<" -  Maiuscolo = "
           <<maius<<endl;
 }
```

Bozza di algoritmo funzione

- Se il parametro formale **c** non contiene una lettera minuscola, restituisci il carattere senza alcuna modifica

- Altrimenti, calcola il corrispondente carattere maiuscolo, sfruttando le proprietà di ordinamento della codifica dei caratteri ASCII:
  - ogni carattere è associato ad un valore intero
    - Che noi assumiamo di **NON CONOSCERE**
  - le lettere da ‘A’ a ‘Z’ sono in ordine alfabetico
  - le lettere da ‘a’ a ‘z’ sono in ordine alfabetico

Funzione

```
/*
 * Dato il carattere in ingresso c restituisce il maiuscolo
 * di c utilizzando solo le proprietà di ordinamento dei
 * codici dei caratteri.
 * Assunzione: se c non è minuscolo, ritorna il
 * carattere inalterato.
 */
char maiuscolo(char c)
{
  if (c<'a' || c>'z')
       return c;
  return c – 'a' + 'A';
}
```

Esercizio 2

Scrivere un programma che stampi a video la tabella dei codici ASCII dei soli caratteri stampabili.

Scrivere il programma supponendo di non conoscere a priori il codice ASCII dei caratteri, ma di sapere solo che i codici dei caratteri stampabili appartengono all'intervallo che va dal codice del carattere ' ' (spazio) al codice del carattere '~'.

Ovviamente non conosciamo neanche il codice dei due estremi, ossia dei caratteri ' ' e '~'.

VINCOLI NELLA PROSSIMA SLIDE

Esercizio 2

VINCOLI

1. Per ogni carattere si stampa il carattere stesso ed il codice ASCII. Formattare la tabella in maniera tale che non occupi più di 25 righe.

2. Implementare la stampa sia mediante due cicli nidificati che mediante un solo ciclo.

3. Provare a scrivere due varianti del programma, utilizzando, rispettivamente una variabile di tipo int ed una variabile di tipo char per memorizzare i codici ASCII.

Esercizio 2

ESEMPIO DI OUTPUT

```
32      33 !    34 "    35 #    36 $    37 %    38 &    39 '
40 (    41 )    42 *    43 +    44 ,    45 -    46 .    47 /
48 0    49 1    50 2    51 3    52 4    53 5    54 6    55 7
56 8    57 9    58 :    59 ;    60 <    61 =    62 >    63 ?
64 @    65 A    66 B    67 C    68 D    69 E    70 F    71 G
72 H    73 I    74 J    75 K    76 L    77 M    78 N    79 O
80 P    81 Q    82 R    83 S    84 T    85 U    86 V    87 W
88 X    89 Y    90 Z    91 [    92 \    93 ]    94 ^    95 _
96 `    97 a    98 b    99 c    100 d   101 e   102 f   103 g
104 h   105 i   106 j   107 k   108 l   109 m   110 n   111 o
112 p   113 q   114 r   115 s   116 t   117 u   118 v   119 w
120 x   121 y   122 z   123 {   124 |   125 }   126 ~
```

Esercizio 3

Scrivere un programma che legge in ingresso un intero e, se il valore inserito appartiene all'insieme dei codici ASCII dei caratteri stampabili, stampa il carattere corrispondente, altrimenti stampa un messaggio di errore.

Scrivere il programma supponendo di non conoscere il codice ASCII dei caratteri, ma di sapere solo che i codici dei caratteri stampabili appartengono all'intervallo che va dal codice del carattere ' ' (spazio) al codice del carattere'~'.

Ovviamente non conosciamo neanche il codice dei due estremi, ossia dei caratteri ' ' e '~'.

SEGUE NELLA PROSSIMA SLIDE

Esercizio 3

Scrivere il programma utilizzando una variabile di tipo int per memorizzare il codice. Per gestire il caso in cui si stampi effettivamente il carattere, realizzare la stampa assegnando tale codice ad una seconda variabile di tipo char e stampando tale variabile.

Assumere che in un assegnamento in cui si ha come lvalue (ossia a sinistra dell'assegnamento) l'indirizzo di una variabile di tipo char, NON SI POSSA avere come rvalue (ossia a destra dell'assegnamento) unvalore di tipo int.

Un possibile output sullo schermo e' il seguente:

```
Inserisci un carattere: M
Il codice corrispondente e': 77
```

Esercizio 4

Scrivere un programma che chieda all'utente di inserire un carattere e stampi a schermo il codice ASCII del carattere inserito

Assumere che in un assegnamento in cui si ha come lvalue (ossia a sinistra dell'assegnamento) l'indirizzo di una variabile di tipo int, NON SI POSSA avere come rvalue (ossia a destra dell'assegnamento) un valore di tipo char.

Un possibile output sullo schermo e' il seguente:

```
Inserisci un carattere: M
Il codice corrispondente e': 77
```

Esercizio 5

Scrivere un programma che legga un carattere da stdin, ne incrementi il valore di 1 e lo stampi su stdout.

Esempi:

```
Immettere un carattere: r
Dopo l'incremento: s

Immettere un carattere: 2
Dopo l'incremento: 3
```

Esercizio 6 (Espressione condizionale)

Senza utilizzare nè l'istruzione **`if`** nè l'istruzione **`switch`** nè le istruzioni cicliche, scrivere un programma che legge un numero intero da stdin e lo memorizza in una variabile n.

Se il numero letto è minore di 0 oppure maggiore di 10, nel programma si assegna rispettivamente il valore 0 oppure 10 ad n. Infine si stampa il valore di n.

Esercizio 7 (visibilità delle variabili)

Scrivere un programma che mostri la visibilità delle variabili. In particolare, nel programma bisogna definire 1) una funzione aggiuntiva oltre la funzione main. Tale funzione deve prendere in ingresso un valore intero 2) una variabile globale di nome x, inizializzata ad un qualche valore 3) una variabile di nome x all'interno di un blocco all'interno della funzione main 4) una variabile di nome x all'interno di un blocco all'interno della funzione aggiuntiva

La funzione main deve stampare il valore della variabile globale x, incrementarlo, e quindi stamparlo di nuovo. Quindi deve stampare il valore della variabile x definita nel blocco interno, incrementarlo e stamparlo di nuovo. Poi deve invocare due volte la funzione aggiuntiva passandogli prima la x definita nel blocco interno, quindi la x globale.

La funzione aggiuntiva deve stampare il valore della variabile globale x, incrementarlo, e quindi stamparlo di nuovo. Quindi deve stampare il valore della variabile x definita nel proprio blocco interno, incrementarlo e stamparlo di nuovo. Assieme al valore della variabile x (globale o definita nel proprio blocco interno), la funzione aggiuntiva deve stampare ogni volta anche il valore del parametro attuale con cui è stata invocata.

Esercizio 7 (visibilità delle variabili)

ESEMPIO DI OUTPUT:

```
Sono il main e la variabile x vale 300
Sono il main e la variabile x vale 301 dopo l'incremento
Sono il main dentro il blocco e la variabile x vale 3
Sono il main dentro il blocco e la variabile x vale 4 dopo l'incremento
Sono il main dentro il blocco e adesso chiamo la funzione
Sono la funzione e la mia x vale 301 mentre la x di chi mi chiama vale 4
Sono la funzione e la mia x vale 302 dopo l'incremento, mentre la x di
chi mi chiama vale 4
Sono la funzione dentro il blocco e la mia x vale 0, mentre la x di chi
mi chiama vale 4
Sono la funzione dentro il blocco e la mia x vale 1 dopo l'incremento,
mentre la x di chi mi chiama vale 4
Sono il main e adesso chiamo la funzione
Sono la funzione e la mia x vale 302 mentre la x di chi mi chiama vale
302
Sono la funzione e la mia x vale 303 dopo l'incremento, mentre la x di
chi mi chiama vale 302
Sono la funzione dentro il blocco e la mia x vale 0, mentre la x di chi
mi chiama vale 302
Sono la funzione dentro il blocco e la mia x vale 1 dopo l'incremento,
mentre la x di chi mi chiama vale 302
```