

Esercizi di verifica

1 – Approfondimento sul linguaggio C: operatori sui singoli bit (*bitwise operators*)

P2_01_01_AC

1. [liv.1] Scrivere una *function C*

```
char low_upp(char ch)
```

che cambia il carattere in input da minuscolo a maiuscolo e viceversa automaticamente mediante gli operatori bitwise.

2. [liv.1] Scrivere una *function C*

```
char rotate(char ch, char n_bit)
```

per ruotare di n bit (n_bit), verso sinistra o verso destra (rispettivamente per n_bit<0 e per n_bit>0), il contenuto di una variabile char mediante gli operatori bitwise.

P2_01_02_AC

3. [liv.1] Scrivere una *function C* che, dopo aver estratto i bit da una variabile intera X (tipo char, short o int) visualizzi i bit e poi ne calcoli il valore corrispondente dalla formula:

$$\text{Val_X} = b[n-1] \cdot 2^{n-1} + \dots + b[2] \cdot 2^2 + b[1] \cdot 2^1 + b[0] \cdot 2^0$$

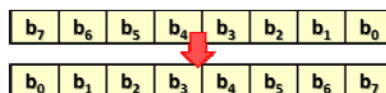
dove b è l'array dei bit di X: b_j, per j=0, 1, ..., n-1 dal meno significativo al più significativo, (dove n=8 per il tipo char, n=16 per il tipo short o n=32 per il tipo int). Confrontare il risultato Val_X con il valore immesso per la variabile intera X dichiarata una volta *signed* ed un'altra *unsigned*.

4. [liv.1] Scrivere una *function C* per estrarre dalla variabile intera X i k bit più significativi (k>0) o meno significativi (k<0), dove X e k sono i parametri di input, usando:

- 1) Una maschera.
- 2) L'operatore di shift (>> o <<).
- 3) Il prodotto o la divisione per potenze di 2.

Scrivere una *function* per ognuna delle metodologie precedenti.

5. [liv.3] Scrivere una *function C* per invertire l'ordine dei bit di una variabile intera A mediante operatori bitwise.



6. [liv.3] Scrivere una *function C* che estragga da una variabile di tipo intero A i suoi bit di posto pari mediante operatori bitwise.

