

Task 3
Bits
Deadline: 06/11/2022 ore 23.59

Scrivere un programma in C++ che legge un numero **rappresentato in complemento a 2** con n bit. Inizialmente il programma deve chiedere il numero n di bit per rappresentare il numero, controllando che $n \geq 2$. Nel caso in cui l'utente inserisca un valore $n < 2$, il programma deve stampare un opportuno messaggio d'errore e chiedere nuovamente l'inserimento del valore di n . Dopodiché, il programma dovrà chiedere all'utente di inserire le cifre del numero rappresentato in complemento a 2 un bit alla volta **e memorizzarle necessariamente in un array**, partendo dal bit più significativo. Il programma dovrà controllare la correttezza della cifra inserita e richiedere l'inserimento della cifra finché non è corretta.

Terminato l'inserimento del numero, il programma dovrà chiedere all'utente una singola operazione che si vuole effettuare sul numero inserito precedentemente, dove:

- 0: stampa a video del numero rappresentato in complemento a 2
- 1: stampa a video del decimale corrispondente al numero inserito
- 2: stampa a video dell'opposto del numero inserito (in binario): nel caso in cui l'opposto non sia rappresentabile in complemento a 2 con n bit, stampare un opportuno messaggio d'errore.

Dopo aver effettuato l'operazione richiesta, il programma termina.

Suggerimenti:

- per l'operazione elevamento a potenza: utilizzare la funzione `pow` della libreria `cmath`
- per il calcolo del decimale corrispondente e dell'opposto (operazioni n° 1 e 2): scorrere l'array da destra verso sinistra

Esempio d'esecuzione (1)

```
Bits: -1
Errore!
Bits: -2
Errore!
Bits: 0
Errore!
Bits: 4
Inserisci il bit nella posizione 3: -1
Errore!
Inserisci il bit nella posizione 3: 2
Errore!
Inserisci il bit nella posizione 3: 1
Inserisci il bit nella posizione 2: 0
Inserisci il bit nella posizione 1: 0
Inserisci il bit nella posizione 0: 0
Operazione:
    0 - Stampa binario:
    1 - Converti in decimale:
    2 - Calcola opposto:
-1
Errore!
5
Errore!
0
Numero binario: 1000
```

Esempio d'esecuzione (2)

```
Bits: 4
Inserisci il bit nella posizione 3: 1
```

Inserisci il bit nella posizione 2: 0
Inserisci il bit nella posizione 1: 0
Inserisci il bit nella posizione 0: 0
Operazione:

- 0 - Stampa binario:
- 1 - Converti in decimale:
- 2 - Calcola opposto:

1

Numero decimale: -8

Esempio d'esecuzione (3)

Bits: 4
Inserisci il bit nella posizione 3: 1
Inserisci il bit nella posizione 2: 0
Inserisci il bit nella posizione 1: 0
Inserisci il bit nella posizione 0: 0
Operazione:

- 0 - Stampa binario:
- 1 - Converti in decimale:
- 2 - Calcola opposto:

2

Overflow!

Esempio d'esecuzione (4)

Bits: 4
Inserisci il bit nella posizione 3: 0
Inserisci il bit nella posizione 2: 1
Inserisci il bit nella posizione 1: 1
Inserisci il bit nella posizione 0: 1
Operazione:

- 0 - Stampa binario:
- 1 - Converti in decimale:
- 2 - Calcola opposto:

0

Numero binario: 0111

Esempio d'esecuzione (5)

Bits: 4
Inserisci il bit nella posizione 3: 0
Inserisci il bit nella posizione 2: 1
Inserisci il bit nella posizione 1: 1
Inserisci il bit nella posizione 0: 1
Operazione:

- 0 - Stampa binario:
- 1 - Converti in decimale:
- 2 - Calcola opposto:

2

Opposto: 1001

L'utilizzo di qualsiasi costrutto/libreria non spiegato a lezione alla data 25/10/2022 comporta il non superamento del task.