

**Cognome:**

**Nome:**

**Matricola:**

2 Punti

1. Dato il numero binario 10101.011:
  - (a) fornire la sua rappresentazione ottale;
  - (b) fornire la sua rappresentazione esadecimale.

4 Punti

2. Mostrando i passaggi fondamentali:
  - (a) rappresentare i numeri decimali 51 e  $-67$  in complemento a due con 8 bit;
  - (b) eseguire la somma algebrica dei numeri ottenuti e riportare (separatamente) il risultato;
  - (c) indicare l'eventuale presenza di overflow e come rilevarla.

4 Punti

3. Mostrando i passaggi fondamentali, ricavare il valore decimale del seguente numero in virgola mobile rappresentato secondo lo standard IEEE 754 a 32 bit: 1 1000001 1100000000000000000000

3 Punti

4. Giustificando in modo preciso la risposta, si dica se le seguenti espressioni booleane sono tra loro equivalenti (ovvero, identificano la stessa funzione delle variabili booleane  $x, y, z$ ):

$$\overline{(x \cdot y) + \bar{z}} \quad \text{e} \quad (\bar{x} + \bar{y}) \cdot z + \bar{y} \cdot z$$

2 Punti

5. Si consideri il seguente programma.

```
1 #include <stdio.h>
2 main(){
3     int x=10;
4     int y=20;
5
6     int f(int a){
7         int x=30;
8         return a+x+y;
9     }
10
11     x=f(10);
12
13     printf("Valore x = %d", x);
14
15     system("pause");
16 }
```

Giustificando la risposta, si dica quale valore viene stampato.

5 Punti

6. Scrivere un programma C che riceva in ingresso un numero intero  $n \geq 0$  e calcoli l' $n$ -esimo elemento della successione definita nel modo seguente:

- il primo elemento (di indice 0)  $a_0$  è 10
- per ogni elemento  $a_i$  con  $i > 0$ , se  $a_{i-1}$  (il precedente) è divisibile per 3 allora  $a_i = a_{i-1}/3+1$  ( $a_i$  è l'elemento precedente diviso per 3 cui si aggiunge 1), altrimenti  $a_i = a_{i-1} + 10$ .

A titolo di esempio, i primi numeri della successione sono 10, 20, 30, 11, 21, ... Quindi, se il programma riceve in ingresso 0 deve stampare 10, se riceve 1 deve stampare 20, se riceve 2 deve stampare 30, se riceve 4 deve stampare 11, ecc. ecc.

10 Punti

7. Il *gioco dell'oca* è un tipico gioco per bambini che include un tabellone con un percorso a caselle (vedi figura). I giocatori lanciano due dadi a turno e, partendo dalla casella 1, spostano un proprio contrassegno in avanti di tante caselle quanti sono i punti corrispondenti alla somma dei dadi ottenuta: vince il giocatore che raggiunge per primo la casella finale.



Si consideri un particolare gioco dell'oca in cui valgono le seguenti regole:

- il percorso è costituito da 100 caselle;
- in alcune caselle è indicato un numero positivo: se un giocatore arriva in una di queste caselle, il contrassegno viene immediatamente spostato in avanti di tante caselle pari al numero indicato (ad esempio, se un giocatore arriva alla casella 90 e in essa è indicato il numero 3, il contrassegno viene automaticamente spostato alla casella 93);
- in alcune caselle è indicato un numero negativo: se un giocatore arriva ad una di queste caselle, il contrassegno viene immediatamente spostato indietro di tante caselle pari al numero indicato (ad esempio, se un giocatore arriva alla casella 90 e in essa è indicato il numero  $-3$ , il contrassegno viene automaticamente spostato alla casella 87);
- anche in seguito ad uno spostamento (vedi punti precedenti) si può arrivare ad una casella con un numero indicato: in questo caso il contrassegno viene spostato nuovamente e così via fino a quando si arriva ad una casella senza indicazione di spostamento;
- per vincere, un giocatore deve arrivare esattamente alla casella 100: se il punteggio è maggiore di 100, si deve tornare indietro di tante caselle quanti sono i punti in eccesso (ad esempio, se un giocatore parte dalla casella 95 e totalizza 7, il suo contrassegno termina nella casella 98 - sempre che in questa casella non sia indicato uno spostamento).

Scrivere un programma C che simuli il gioco dell'oca tra due giocatori:

- Prima di tutto, il programma acquisisce da tastiera *per ogni casella* il relativo spostamento (0 indica nessun spostamento, un numero positivo indica uno spostamento in avanti, un numero negativo indica uno spostamento all'indietro). Il programma deve verificare che lo spostamento non porti al di fuori delle caselle del percorso, altrimenti deve ripeterne l'acquisizione. Gli spostamenti devono essere memorizzati in un opportuno vettore.
- Partendo dal giocatore 1, il programma chiede poi a turno ai giocatori il numero ottenuto dal lancio dei dadi, ripetendo l'acquisizione se il numero non è compreso tra 2 e 12: dopo ogni acquisizione, calcola e stampa la casella di arrivo finale.
- Il programma termina quando uno dei due giocatori arriva alla casella 100, stampando il vincitore (giocatore 1 o giocatore 2).