

Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_ Matr: \_\_\_\_\_ Postazione: \_\_\_\_\_

Scrivere un programma in linguaggio C (chiamare il progetto con la propria **<matricola>**) che abbia il comportamento descritto nel seguito. Il tempo a disposizione è di **2 ore**. Al termine del tempo gli elaborati salvati su **U:\** verranno raccolti automaticamente dal sistema. Eventuali documenti sono disponibili in **T:\Bertozzi**, usare **wordpad** per leggere eventuali file di testo.

Si vuole sviluppare una specie di animazione in cui una pallina (simulata usando il carattere **O**) si muove all'interno di un rettangolo rimbalzando sulle pareti. In particolare la pallina si muove di una sola posizione alla volta in "diagonale" rimbalzando sulle pareti con un angolo di 90°.

L'utente ha la possibilità di modificare la direzione dello spostamento della **O** usando i tasti 'Q' e 'E' che modificano la direzione della **O** di 90 gradi; in senso antiorario la 'Q', in senso orario la 'E'.

L'aggiornamento della posizione della **O** avviene ogni 300 ms.

Ai fini del progetto, nello scheletro di codice fornito, sono già definite e usabili le seguenti funzioni:

- **char getcommand(void)**: restituisce il codice ASCII del tasto premuto oppure 0 se non è stato premuto nessun tasto.
- **void gotoxy(int c, int r)**: permette di impostare la colonna c e la riga r per la prossima scrittura. Usata prima di una **printf()** o similari, permette di scrivere in una posizione arbitraria della console. Si tenga presente che la posizione in alto a sinistra è quella di coordinate (0,0). L'indice di colonna aumenta verso destra e quello di riga verso il basso.

Per lo sviluppo del codice:

1. (10) Si definisca una funzione **void printfield(void)**; che stampa sullo schermo il rettangolo in cui si muove la pallina di 23 righe e 97 colonne, delimitato da un bordo. Per il bordo, si usino i caratteri '+' per gli angoli, il '-' per i bordi orizzontali e il '|' (ASCII 124) per quelli verticali. La si invochi dalla **main()**;
2. (5) Si definisca opportuna struttura dati per memorizzare lo stato della pallina, quali posizione e direzione di movimento. La si usi per inizializzare la posizione di una pallina nel rettangolo con coordinate e direzione iniziale casuale. Si definiscano quindi le funzioni:
  - a. (1) **void printball(?)**; che stampa la **O** sullo schermo
  - b. (1) **void eraseball(?)**; che cancella la **O** dallo schermo sovrascrivendola con uno spazio
  - c. (6) **void updaball(?)**; che aggiorna i dati della pallina in base alla direzione di spostamento e a eventuali rimbalzi
3. (4) Si usi, nella **main()**, un ciclo da cui si esce solamente premendo il tasto 'X'. All'interno del ciclo si usino le funzioni definite al punto precedente per simulare l'animazione. Il ritardo nell'aggiornamento dello schermo è ottenibile usando la funzione **int usleep(usec)**; che si blocca per usec µs (nota: 1 ms = 1000 µs)
4. (6) Si modifichi la **updaball()** o il ciclo al punto #3 per gestire la pressione dei tasti 'Q' ed 'E'.

**Il codice va sviluppato nell'ordine indicato. Per ogni punto implementato si verifichi il corretto funzionamento. La correzione termina al primo punto non implementato correttamente.**