

Prova d'Esame

Ingegneria e Scienze Informatiche A.A. 2022-2023

Si vuole implementare in linguaggio C una libreria di funzioni per l'implementazione di una versione semplificata del Blackjack. Si deve consegnare un file **blackjack.c** contenente l'implementazione delle funzioni che si trovano nel file header **blackjack.h**.

Per modellare i concetti principali del gioco sono state definite le seguenti strutture ed enumerazioni:

```
enum suits { HEARTS, DIAMONDS, CLUBS, SPADES};

/*Modella una carta*/
struct single_card {
    int ranking;
    enum suits suit;
};

/*Modella un insieme di carte*/
struct cards {
    struct single_card* cards;
    int size;
};
```

Le funzioni da implementare sono le seguenti:

- **deck_gen (4 punti):** nel Blackjack viene utilizzato un mazzo composto da più mazzi standard. Ciascun mazzo standard è composto da 4 serie di carte (una per seme), che vanno da **A** a **K**, Il numero della carta (ranking) è un intero che va da **1** a **13**, dove 1 è l'asso e 13 è il Re.

MAZZO STANDARD

HEARTS ♥												
A♥	2♥	3♥	4♥	5♥	6♥	7♥	8♥	9♥	10♥	J♥	Q♥	K♥
DIAMONDS ♦												
A♦	2♦	3♦	4♦	5♦	6♦	7♦	8♦	9♦	10♦	J♦	Q♦	K♦
CLUBS ♣												
A♣	2♣	3♣	4♣	5♣	6♣	7♣	8♣	9♣	10♣	J♣	Q♣	K♣
SPADES ♠												
A♠	2♠	3♠	4♠	5♠	6♠	7♠	8♠	9♠	10♠	J♠	Q♠	K♠

La funzione `deck_gen` prende in input il numero di mazzi standard da cui è composto il mazzo di gioco e:

- crea un mazzo (istanza di `struct cards`);
- alloca la memoria **contigua** necessaria; (`CARDSxDECK * num_decks`)
- popola il mazzo generato.

NB: `CARDSxDECK` rappresenta il numero di carte in un mazzo standard, ed è definito in `blackjack.h`

Esempio

Con `num_decks = 2` il risultato è il seguente:

```
Stampa mazzo generato:
A♥ 2♥ 3♥ 4♥ 5♥ 6♥ 7♥ 8♥ 9♥ 10♥ J♥ Q♥ K♥
A♦ 2♦ 3♦ 4♦ 5♦ 6♦ 7♦ 8♦ 9♦ 10♦ J♦ Q♦ K♦
A♣ 2♣ 3♣ 4♣ 5♣ 6♣ 7♣ 8♣ 9♣ 10♣ J♣ Q♣ K♣
A♠ 2♠ 3♠ 4♠ 5♠ 6♠ 7♠ 8♠ 9♠ 10♠ J♠ Q♠ K♠
A♥ 2♥ 3♥ 4♥ 5♥ 6♥ 7♥ 8♥ 9♥ 10♥ J♥ Q♥ K♥
A♦ 2♦ 3♦ 4♦ 5♦ 6♦ 7♦ 8♦ 9♦ 10♦ J♦ Q♦ K♦
A♣ 2♣ 3♣ 4♣ 5♣ 6♣ 7♣ 8♣ 9♣ 10♣ J♣ Q♣ K♣
A♠ 2♠ 3♠ 4♠ 5♠ 6♠ 7♠ 8♠ 9♠ 10♠ J♠ Q♠ K♠
```

La signature della funzione è:

```
struct cards deck_gen(int num_decks);
```

- **deck_shuffle (4 punti):** prende in input un puntatore al mazzo di carte e lo mischia

Esempio

Risultato di una chiamata a `deck_shuffle`, a cui è stato dato in input il mazzo dell'esempio precedente:

```
Stampa mazzo mischiato:
A♥ A♠ A♣ 4♥ 4♥ A♦ 5♥ 8♣ Q♠ K♠ J♥ Q♥ A♥
10♠ 9♦ 3♦ 2♥ A♠ 6♠ K♠ 8♦ 2♠ 4♠ 2♦ Q♦ K♦
6♦ 7♦ 4♠ 3♠ 5♠ 10♦ 7♠ 2♠ 3♥ 9♠ J♠ 10♠ 7♦
6♥ 9♥ 10♥ 8♠ 5♠ J♠ K♠ 6♥ 2♠ 6♠ J♦ 10♦ 4♦
3♠ 8♠ 3♥ 5♥ 7♥ K♥ Q♠ 7♥ 8♠ 8♦ J♥ Q♥ K♦
7♠ 2♦ 9♦ 2♥ 6♦ 6♠ 10♠ J♠ 4♠ Q♠ J♦ Q♦ K♥
A♠ 3♦ 4♦ 9♠ 5♠ 5♠ A♦ 8♥ 9♠ 10♠ 10♥ 5♦ Q♠
3♠ 5♦ 7♠ 4♠ 7♠ 3♠ 2♠ 9♥ 9♠ 8♥ J♠ 6♠ K♠
```

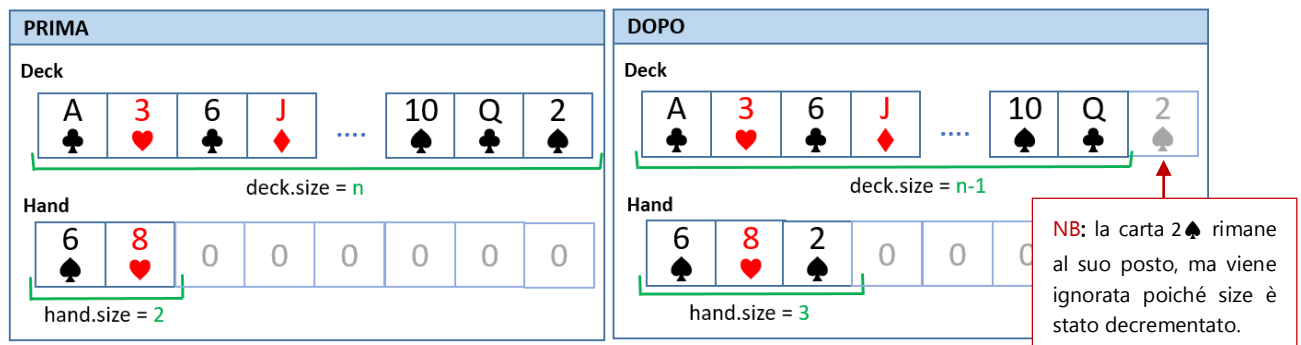
La signature della funzione è la seguente:

```
void deck_shuffle(struct cards* deck);
```

⚠ **Attenzione:** il seed è già stato inizializzato nel main mediante `srand`. Non è necessario farlo anche in questa funzione.

- **deal_card (4 punti):** prende in input due puntatori: uno al mazzo e uno alla mano del giocatore. Distribuisce al player una carta presa dal mazzo.

Distribuire una carta significa prendere l'ultimo elemento del mazzo e aggiungerlo alla mano del player, come mostrato in figura



La signature della funzione è:

```
void deal_card(struct cards* deck, struct cards* hand)
```

- **sum_points (4 punti):** prende in input la mano di un giocatore e ne calcola il **punteggio**. Il punteggio corrente di un giocatore si calcola facendo la somma dei valori delle carte che ha in mano. Il valore di una carta si calcola nel seguente modo:
 - J, Q e K valgono **10 punti**.
 - 2, 3, ..., 10 mantengono il loro valore nominale (**ranking**)
 - A assume il valore **1** o **11** a seconda dei seguenti casi:
 - **11** se punteggio totale **non** supera **21**
 - **1** altrimenti

La signature della funzione è la seguente:

```
int sum_points(struct cards hand);
```