

---

# Esercizi con puntatori

Filippo Cugini

---

## Esempio: il poker

---

Realizziamo un programma che distribuisca una mano di poker di cinque carte

Poi scrivete ulteriori funzioni per determinare se tra le cinque carte c'è:

- a) una coppia
- b) due coppie
- c) tre dello stesso tipo (es. tre fanti)
- d) quattro dello stesso tipo (es. quattro assi)
- e) un colore solo (tutte le carte dello stesso seme)
- f) una scala (valori in sequenza)

# Esempio: il poker

---

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define SUITS 4
#define FACES 13
#define CARDS 52
```

# Esempio: il poker

```
// prototipi di funzione
void shuffle(unsigned int deck[][FACES]);
void deal(unsigned int deck[][FACES],
          unsigned int hand[][2],
          char *suit[], char *face[]);
void pair(unsigned int hand[][2], char *suit[],
          char *face[]);
void threeOfKind(unsigned int hand[][2],
                 char *suit[], char *face[]);
void fourOfKind(unsigned int hand[][2],
                char *suit[], char *face[]);
void straightHand(unsigned int hand[][2],
                  char *suit[], char *face[]);
void flushHand(unsigned int hand[][2],
               char *suit[], char *face[]);
```

## Esempio: il poker

```
int main(void)
{
    char * suit[SUITS] = { "Hearts", "Diamonds",
                           "Clubs", "Spades" };

    char * face[FACES] = { "Ace", "Deuce", "Three",
                           "Four", "Five", "Six", "Seven", "Eight",
                           "Nine", "Ten", "Jack", "Queen", "King" };

    unsigned int deck[SUITS][FACES]; //mazzo carte

    unsigned int hand[5][2]; // una mano di gioco
```

# Esempio: il poker

```
unsigned int deck[SUITS][FACES]; //mazzo carte
```

		Asso	Due	Tre	Quattro	Cinque	Sei	Sette	Otto	Nove	Dieci	Fante	Regina	Re
		0	1	2	3	4	5	6	7	8	9	10	11	12
Cuori	0													
Quadri	1													
Fiori	2													
Picche	3													

deck[2][12] rappresenta il Re di Fiori

Fiori      Re

## Esempio: il poker

```
// effettua un ciclo sulle righe del mazzo
for (size_t row = 0; row < SUITS; ++row)
{
    // effettua un ciclo sulle colonne
    for (size_t col = 0; col < FACES; ++col)
    {
        deck[row][col] = 0; // inizializza a 0
    }
}

srand(time(NULL)); // seme per numeri casuali

// mischia e distribuisce una mano di 5 carte
shuffle(deck);
deal(deck, hand, suit, face);
```

## Esempio: il poker

---

```
// determina il valore del poker della mano  
  
pair(hand, suit, face);  
  
threeOfKind(hand, suit, face);  
  
fourOfKind(hand, suit, face);  
  
straightHand(hand, suit, face);  
  
flushHand(hand, suit, face);  
}
```



# Esempio: il poker

```
// mischia il mazzo
void shuffle(unsigned int deck[][FACES]) {
    size_t row; // valore seme della carta
    size_t col; // valore figura della carta
    // scegli un elemento del mazzo a caso
    for (unsigned int card=1; card<=CARDS; ++card)
    {
        do { // scegli nuova posizione finche' non si
            // trova un elemento non occupato
            row = rand() % SUITS;
            col = rand() % FACES;
        } while(deck[row][column] != 0);
        // inserisci la carta
        deck[row][column] = card;
    }
}
```

## Esempio: il poker

```
// distribuisce una mano di poker
void deal(unsigned int deck[][FACES], unsigned int
          hand[][2], char *suit[], char *face[])
{
    // contatore per la posizione nella mano
    unsigned int r = 0;
    puts("The hand is:\n");
    // ripeti per distribuire le carte
```

## Esempio: il poker

```
for (size_t card = 1; card < 6; ++card) {
    for (size_t row = 0; row < SUITS; ++row) {
        for (size_t col = 0; col < FACES; ++col) {
            if (deck[row][col] == card) {
                hand[r][0] = row;
                hand[r][1] = col;
                printf("%5s of %-8s\n", face[col],
                    suit[row]);
                ++r;
            }
        }
    }
    puts("\n");
}
```

## Esempio: il poker (seconda parte)

```
// determina se ci sono coppie nella mano
void pair(unsigned int hand[][2], char *suit[],
          char *face[])
{
    // contatore di carte di ciascun tipo nella mano
    unsigned int counter[FACES] = {0};
    for (size_t r = 0; r < 5; ++r) {
        ++counter[ hand[r][1] ];
    }
    // stampa il risultato se c'e' una coppia
    for (size_t p = 0; p < FACES; ++p) {
        if (counter[p] == 2) {
            printf("Pair of %ss.\n", face[p]);
        }
    }
}
```

## Esempio: il poker (seconda parte)

---

```
// determina se ce ne sono tre di un tipo nella mano
void threeOfKind(unsigned int hand[][2], char *suit[],
                char *face[])
{
    // contatore di carte di ciascun tipo
    unsigned int counter[FACES] = { 0 };
    for (size_t r = 0; r < 5; ++r) {
        ++counter[hand[r][1]];
    }
    // stampa il risultato se ce ne sono tre di un tipo
    for (size_t t = 0; t < FACES; ++t) {
        if (counter[t] == 3) {
            printf("Three %ss.\n", face[t]);
        }
    }
}
```

## Esempio: il poker (seconda parte)

---

```
// determina se ci sono quattro di un tipo nella mano
void fourOfKind(unsigned int hand[][2], char *suit[],
               char *face[])
{
    // contatore di carte di ciascun tipo
    unsigned int counter[FACES] = { 0 };
    for (size_t r = 0; r < 5; ++r) {
        ++counter[hand[r][1]];
    }
    // stampa il risultato se c'è poker
    for (size_t k = 0; k < FACES; ++k) {
        if (counter[k] == 4) {
            printf("Four %ss.\n", face[k]);
        }
    }
}
```

## Esempio: il poker (seconda parte)

---

```
// determina se c'e' una scala nella mano
void straightHand(unsigned int hand[][2],
                  char *suit[], char *face[])
{
    // array che mantiene una copia della mano
    unsigned int s[5] = { 0 };
    unsigned int temp; // intero temporaneo
    // copia le posizioni delle colonne
    // per effettuare l'ordinamento
    for (size_t r = 0; r < 5; ++r)
        s[r] = hand[r][1];
}
```

# Esempio: il poker (seconda parte)

---

```
// bubble sort per le posizioni delle colonne
for (size_t pass = 1; pass < 5; ++pass) {
    for (size_t comp = 0; comp < 4; ++comp) {
        if (s[comp] > s[comp + 1])
        {
            temp = s[comp];
            s[comp] = s[comp + 1];
            s[comp + 1] = temp;
        }
    }
}
```



## Esempio: il poker (seconda parte)

---

```
// controlla se le colonne ordinate sono una scala
if (s[4] - 1 == s[3] && s[3] - 1 == s[2]
    && s[2] - 1 == s[1] && s[1] - 1 == s[0])
{
    printf("Straight from %s to %s.\n",
          face[s[0]], face[s[4]]);
}
}
```

## Esempio: il poker (seconda parte)

```
// determina se c'e' un colore nella mano
void flushHand(unsigned int hand[][2], char *suit[],
               char *face[])
{
    // contatore di carte di ciascun tipo
    unsigned int count[SUITS] = { 0 };
    for (size_t r = 0; r < 5; ++r) {
        ++count[hand[r][0]];
    }
    for (size_t f = 0; f < SUITS; ++f) {
        if (count[f] == 5) {
            printf("Flush of %ss.\n", suit[f]);
        }
    }
}
```

## Esempio: il poker (seconda parte)

---

### Esercizio consigliato

Usate le funzioni sviluppate per scrivere un programma che:

- distribuisca due mani di poker di cinque carte,
- e determini qual è la mano migliore

## Esercizio

Cosa fa questo programma, supponendo che l'utente inserisca due stringhe di uguale lunghezza?

# Uso di stringhe

---

```
#include <stdio.h>
#define SIZE 80

void mystery1(char *s1, const char *s2);
// prototipo

int main(void)
{
    char string1[SIZE]; // crea un array di char
    char string2[SIZE]; // crea un array di char
    puts("Enter two strings: ");
    scanf("%79s%79s" , string1, string2);
    mystery1(string1, string2);
    printf("%s", string1);
}
```

```
void mystery1(char *s1, const char *s2)
{
    while (*s1 != '\0') {
        ++s1;
    }

    for (; *s1 = *s2; ++s1, ++s2) {
        ; // istruzione vuota
    }
}
```

Risposta:

Concatena stringhe

```
Enter two strings: string1 string2  
string1string2
```