

## Capitolo 9 - Strutture (struct)

### 1. La struttura "punto"

Definisci una struttura adatta a rappresentare un punto nel piano, caratterizzato da coordinata x e coordinata y (entrambe di tipo float). Definisci poi due funzioni, chiamate distanza e puntoMedio. La prima riceve come argomenti due punti e ne restituisce la distanza. La seconda riceve come argomenti due punti e ne restituisce il punto medio. Si ricorda che, dati due punti  $Ax_Ay_A$  e  $Bx_By_B$

- la loro distanza è  $\sqrt{x_1 - x_2^2 + y_1 - y_2^2}$
- il loro punto medio è il punto C di coordinate  $x_C = \frac{x_A + x_B}{2}$  e  $y_C = \frac{y_A + y_B}{2}$

Scrivi infine un programma che richiede all'utente le coordinate di due punti e ne calcola distanza e punto medio utilizzando le funzioni.

---

### 2. Colori RGB

Utilizzando il formato RGB è possibile rappresentare qualsiasi colore tramite tre numeri interi compresi tra 0 e 255 (componente rossa, verde e blu). Un colore può essere reso più chiaro o più scuro aumentando o diminuendo contemporaneamente le tre componenti di una stessa percentuale. Ad esempio, partendo dal colore **(200,100,0)** e aumentando tutte le componenti del 30% si ottiene il colore **(255,130,0)**. Dopo aver definito opportunamente la struttura "colore", scrivi le seguenti funzioni:

- la funzione **inputColore** che richiede all'utente di specificare un colore e restituisce il colore acquisito
  - la funzione **void stampa(colore c)** che stampa le componenti del colore c nel formato (r,g,b)
  - la funzione **nuovoColore**, che riceve come argomenti un colore e una percentuale (positiva o negativa), e restituisce il nuovo colore ottenuto schiarendo o scurendo il colore di input. Attenzione: ciascuna componente di un colore deve rimanere nell'intervallo 0..255. Ad esempio, se si aumenta del 10% la "luminosità" del colore (200,250,0) si dovrà ottenere (220,255,0) e non (220,275,0) poichè il valore 275 non è accettabile nel formato RGB.
- 

### 3. La struttura "data"

Definisci una struttura adatta a rappresentare una data, espressa tramite giorno, mese e anno (tre numeri interi). Scrivi poi le seguenti funzioni:

- int valida(data d);** Verifica se una data è valida (cioè esiste nel calendario). Esempio: 31/11/2015 non è valida perchè novembre ha 30 giorni. La funzione restituisce 1 se l'anno, il mese e il giorno del parametro d costituiscono una data valida, 0 altrimenti.

- b. **data successiva(data d);** Restituisce la data immediatamente successiva alla data d (cioè la data del giorno seguente). Esempio: se d = 31/12/2015, la data successiva è 1/1/2016.
- c. **int confronta(data d1, data d2);** Stabilisce se la data d1 è precedente o successiva rispetto alla data d2. Restituisce -1 se d1 è precedente a d2, 0 se le d1 e d2 sono uguali, +1 se d1 è successiva a d2.
- d. **data aggiungiGiorni(data d, int n );** Restituisce la data che si trova a una distanza di n giorni dalla data d.
- e. **void stampa(data d);** Visualizza la data sull'output standard, nel formato gg/mm/aaaa.
- f. **int differenzaGiorni( data d1, data d2 );** Restituisce il numero di giorni che intercorrono da d1 a d2. Se d2 è precedente a d1, il risultato deve essere negativo. Esempi: d1 = 31/12/2015 d2 = 2/1/2016 differenza = 2 giorni. d1 = 2/1/2015 d2 = 31/12/2015 differenza = -2 giorni.

Si utilizzi la seguente funzione per stabilire se un anno è bisestile:

```
int bisestile(int anno) {
    if (anno % 400 == 0 || (anno % 100 != 0 && anno % 4 == 0))
        return 1;
    else
        return 0;
}
```

---

## Puntatori a struttura

### 4. La struttura "frazione"

Definisci una struttura adatta a rappresentare una frazione, caratterizzata da numeratore e denominatore (due numeri interi, con denominatore sempre diverso da zero). Scrivi poi le seguenti funzioni:

- a. **void acquisisci(frazione\* f);** Chiede all'utente i valori di numeratore e denominatore (con controllo del denominatore) e imposta i campi della struttura ricevuta per indirizzo
- b. **void semplifica(frazione\* a);** Semplifica la frazione ricevuta per indirizzo, cioè divide sia numeratore che denominatore per il loro massimo comun divisore (vedi algoritmo per MCD sotto).
- c. **void stampa(frazione f);** Visualizza la frazione nel formato num/den con i seguenti accorgimenti: 1) se il denominatore è 1, visualizza solo il numeratore. 2) il segno, se presente, è posto sempre al numeratore.
- d. **frazione moltiplica(frazione a, frazione b);** Restituisce nuova la frazione a\*b. Suggerimento: per calcolare il prodotto  $(N1/D1) * (N2/D2)$  conviene prima costruire la frazione  $(N1 * N2) / (D1 * D2)$  e poi semplificarla con la funzione definita in precedenza.
- e. **frazione dividi(frazione a, frazione b);** Restituisce la nuova frazione a/b
- f. **frazione somma(frazione a, frazione b);** Restituisce la nuova frazione a+b. Suggerimento: per calcolare la somma  $N1/D1 + N2/D2$  conviene prima costruire la frazione  $(N1 * D2 + N2 * D1) / (D1 * D2)$ , e poi semplificarla con la funzione definita in precedenza.

- g. `frazione sottrai(frazione a, frazione b)`; Restituisce la nuova frazione  $a-b$

Scrivi un programma che richiede all'utente due frazioni e mostra il risultato delle 4 operazioni. Esempio:

```
Inserisci la frazione A:
Numeratore: 1
Denominatore (non nullo): 2
Inserisci la frazione B:
Numeratore: 2
Denominatore (non nullo): -3
A+B = -1/6
A-B = 7/6
A*B = -1/3
A/B = -3/4
```

Algoritmo per il calcolo del massimo comun divisore:

```
int MCD(int a, int b) {
    if (a<0) a=-a;
    if (b<0) b=-b;
    if (b == 0) return a;
    int r = a % b;
    while(r != 0) {
        a = b;
        b = r;
        r = a % b;
    }
    return b;
}
```

---

## Array di strutture

### 5. Rilevazioni atmosferiche

Una rilevazione atmosferica è caratterizzata da una data (giorno,mese,anno) e dai valori di temperatura e umidità. Utilizzando opportunamente le strutture, scrivi un programma che richiede all'utente i dati relativi a N rilevazioni e mostra le date in cui si sono registrate, rispettivamente, la temperatura massima e l'umidità massima. Esempio:

```
Quante rilevazioni vuoi inserire? 2

Inserisci la rilevazione 1:
Data: 15 8 2015
Temperatura: 29.5
Umidita': 75

Inserisci la rilevazione 2:
Data: 29 7 2015
Temperatura: 31.5
Umidità: 71
```

TEMPERATURA MASSIMA registrata il giorno 29/7/2015 (31.5 gradi)

UMIDITA MASSIMA registrata il giorno 15/08/2015 (75%)

## 6. Carte da gioco

Una carta da gioco può essere rappresentata tramite due numeri interi: il primo numero indica il seme ( 0=cuori, 1=quadri, 2=fiori, 3=picche) e il secondo numero indica il valore ( 1=asso, 2, 3, 4, ..., 9, 11=fante, 12=regina, 13=re ). Escludendo i jolly, quindi, un mazzo è composto da  $13 \times 4 = 52$  carte. E' richiesto di:

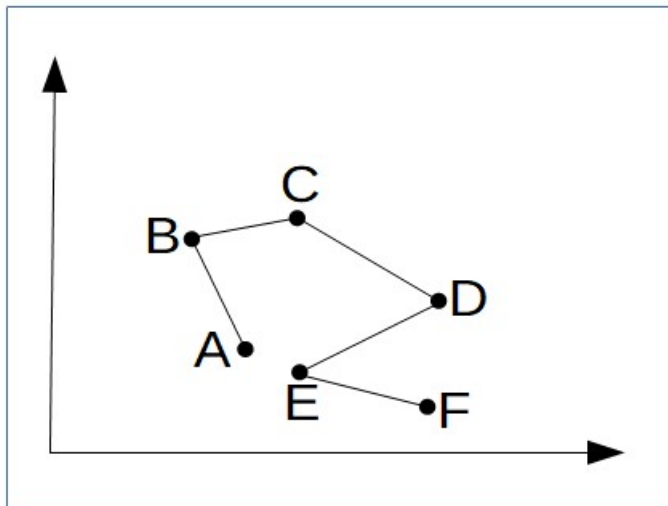
- definire la struttura "carta", composta dai campi seme e valore di tipo intero
- definire la funzione `void stampa(cart a c)` che stampa la descrizione della carta ("7 di picche", "asso di cuori", ecc.)
- definire una funzione che stampa il contenuto di un mazzo di n carte (non necessariamente un mazzo completo)
- generare un array di 52 carte corrispondente ad un mazzo completo e ordinato, e stamparlo. Nota: per convenzione in un mazzo ordinato compaiono prima tutte le carte di cuori, poi quadri, poi fiori e infine picche.
- definire una funzione che "mescola" un mazzo di n carte (non necessariamente un mazzo completo). Per mescolare il mazzo si procede nel modo seguente: per 100 volte si scelgono due indici casuali dell'array e si scambiano le carte nelle posizioni specificate dai due indici.

Utilizzando le funzioni, scrivi un programma che genera un mazzo ordinato, lo stampa, lo mescola, e infine stampa il mazzo mescolato.

## 7. Linea spezzata

Data la struttura:

```
typedef struct {  
    char nome[21];  
    float x;  
    float y;  
} punto;
```



definisci:

- la funzione `punto input()` che richiede all'utente il nome e le coordinate di un punto, e restituisce il punto acquisito
- la funzione `void stampa(punto p)` che stampa il punto nel formato `nome(x,y)` . Esempio: `"A(1,2)"`.
- una funzione chiamata `lunghezzaTotale` che riceve un array di punti e restituisce la lunghezza totale della linea spezzata che congiunge tutti i punti (ved figura)
- una funzione chiamata `distanzaMinima`, che riceve un array di punti e restituisce la distanza tra i due punti più vicini. Nota: i due punti più vicini non sono necessariamente consecutivi: nella figura, ad esempio, i due punti più vicini sono i punti A ed E, e la funzione dovrà restituire la loro distanza.
- la funzione `void puntiPiuVicini(punto linea[], int n, punto* p1, punto* p2)`, che permette di ottenere la coppia di punti (p1,p2) con la minima distanza reciproca. Nota: la distanza fra questi due punti coinciderà con la distanza minima calcolata dalla funzione precedente.

Scrivi infine un programma che consenta di testare le due funzioni su un elenco di punti fornito dall'utente.

---

## 8. Studenti

Definisci una struttura adatta a contenere i dati relativi ad uno studente, sapendo che per ogni studente ci interessano i seguenti dati: matricola, nome, cognome, classe, anno di nascita e sesso. Nota: la matricola è un numero intero che identifica uno studente.

Scrivi poi un programma che consenta, tramite un menu di scelta, di effettuare le seguenti operazioni:

- inserire i dati di nuovi studenti, fino al massimo consentito di 1000 studenti
  - visualizzare la lista degli studenti inseriti fino ad ora
  - visualizzare la lista degli studenti nati in un determinato anno (richiesto all'utente)
  - visualizzare i dati di uno studente di cui si conosce la matricola
  - visualizzare i dati di uno studente di cui si conosce il cognome
  - cancellare uno studente fornendo il numero di matricola
  - visualizzare una tabella che mostra, per ogni classe, il numero di maschi e il numero di femmine. Supporre che il numero di classi diverse sia sicuramente inferiore a 100.
- 

## 9. Torneo di calcio

Due squadre di calcio (squadra A e squadra B) decidono di giocare un "torneo" costituito da 3 partite. Per gestire il torneo, si vuole scrivere un programma che consenta di registrare i goal fatti da ciascun giocatore. Prima di registrare i goal, occorre registrare i giocatori specificando per ognuno: numero identificato (un numero unico all'interno del torneo), nome, cognome, squadra (le uniche due opzioni sono 'A' oppure 'B'). Per ogni goal occorre registrare: numero della partita (1,2 o 3), numero del giocatore che ha effettuato il goal, minuto a cui è avvenuto il goal. Dopo aver definito le strutture necessarie, scrivi un programma che consenta, tramite un menu, di effettuare le seguenti operazioni:

- registrare nuovi giocatori

- b. registrare nuovi goal
  - c. visualizzare i dati del giocatore che ha effettuato il maggior numero di goal
  - d. visualizzare il risultato di ogni partita (numero totale di goal della squadra A e della squadra B) e il nome della squadra vincitrice del torneo
- 

## 10. Birreria

Una birreria propone ai propri clienti una lista con decine di birre di ogni tipo (MAX 50). Di ciascuna birra si vuole memorizzare:

- nome della birra (univoco)
- tipo, espresso da un numero intero (1=Chiara, 2=Scura, 3=Rossa, 4=Ale, 5=Doppio malto, 6=Weiss)
- prezzo di una bottiglia da 33 cl

Richiedere in ordine qualsiasi i dati delle birre in vendita (i nomi delle birre dati in input vanno convertiti in maiuscolo). Successivamente ordinarle in ordine crescente di prezzo. Mostrare un menù CICLICO con le seguenti opzioni:

- A) Stampare l'elenco delle birre (nome, prezzo, descrizione del tipo).
  - B) Visualizzare i dati della birra più cara e di quella meno cara.
  - C) Dato in input il nome di una birra, visualizzare il tipo e il prezzo della bottiglia da 33cl.
  - D) Per ciascuno dei 6 tipi di birra, contare il numero di birre che ci sono nella lista della birreria e stampare i conteggi.
-