

APPELLO SCRITTO DI PROGRAMMAZIONE 1
CORSO DI LAUREA IN MATEMATICA
UNIVERSITÀ DEGLI STUDI DI MILANO
2015–2016
22.II.2017

VINCENZO MARRA

INDICE

Esercizio 1	1
Strutture dati e prime funzioni.	1
Punti: 8.	2
Esercizio 2	2
Menu e allocazione della memoria.	2
Punti: 12.	2
Esercizio 3	3
Calcolo del perimetro medio e del rettangolo di area massima.	3
Punti: 10.	3
<i>Istruzioni per la consegna</i>	5

Avvertenza. Il tema d'esame richiede lo sviluppo di un singolo programma che esegua semplici operazioni su rettangoli nel piano cartesiano. Svilupperete delle funzioni ausiliarie nel risolvere gli esercizi. Integrerete poi tutte le funzioni ausiliarie in un unico programma finale, che sarà costituito da un solo file sorgente. (Il tema d'esame richiede esplicitamente l'implementazione di alcune funzioni ausiliarie. È comunque sempre ammissibile aggiungere altre funzioni ausiliarie qualora lo riteniate opportuno per una migliore implementazione della vostra soluzione.) Si raccomanda di leggere interamente il tema d'esame con attenzione prima di cominciare a scrivere le soluzioni dei singoli esercizi, in modo da avere chiaro l'obiettivo finale.

ESERCIZIO 1

Strutture dati e prime funzioni.

Ultima revisione: 22 febbraio 2017.

Punti: 8.

Usando `typedef` e `struct`, dichiarate un tipo `punto` i cui valori rappresentino punti nel piano cartesiano; i campi ascissa e ordinata devono avere tipo `double`. Sempre usando `typedef` e `struct`, dichiarate poi un tipo `rett` i cui valori rappresentino rettangoli nel piano cartesiano con i lati paralleli agli assi coordinati. Codificate un rettangolo tramite due punti del piano: il suo vertice in posizione sud-ovest, e il suo vertice in posizione nord-est.

Dichiarate e implementate la funzione ausiliaria:

```
void stampa(rett*)
```

che, ricevuto in ingresso un puntatore a un rettangolo, stampi a video il rettangolo nella forma:

```
{(x,y),(x',y')}
```

dove (x,y) sono le coordinate del vertice in posizione sud-ovest del rettangolo, e (x',y') quelle del suo vertice in posizione nord-est. Per esempio, il quadrato $[0,1]^2 \subseteq \mathbb{R}^2$ sarebbe visualizzato così:

```
{(0,0),(1,1)}
```

Dichiarate e implementate anche le funzioni ausiliari seguenti:

- `double base(rett*)`
- `double alt(rett*)`
- `double per(rett*)`
- `double area(rett*)`

Esse, ricevuto in ingresso un puntatore a un rettangolo, restituiscono rispettivamente la base, l'altezza, il perimetro e l'area del rettangolo.

Testate la vostra implementazione prima di proseguire.

ESERCIZIO 2

Menu e allocazione della memoria.**Punti: 12.**

Scrivete una funzione `main` che implementi il menu seguente:

1. Aggiungi rettangolo.
2. Rimuovi rettangolo.
3. Visualizza.
4. Esci.

Dichiarate un variabile globale puntatore di nome `lista` e di tipo `rett*`, inizializzata a `NULL`, che punterà per tutta l'esecuzione del programma a una lista di rettangoli inserita dall'utente. L'utente modifica la lista tramite le opzioni 2 e 3, aggiungendo un nuovo rettangolo alla lista o eliminando l'ultimo rettangolo aggiunto dalla lista. Inizialmente la lista è vuota, il che corrisponde al fatto che il puntatore `lista` vale `NULL`. Usate una variabile locale al `main` di nome `n` e di tipo `int` per memorizzare, in ogni momento dell'esecuzione, la lunghezza della lista corrente.

Se l'utente digita una stringa che non corrisponde a un'opzione esistente, il programma visualizza un messaggio d'errore appropriato e torna al menu.

Se l'utente sceglie 4, il programma termina.

```
Vincenzos-MacBook-Air:Soluzione enzo$ ./a.out
1.   Aggiungi rettangolo.
2.   Rimuovi rettangolo.
3.   Visualizza.
4.   Esci.
3
Nessun rettangolo in memoria.
1.   Aggiungi rettangolo.
2.   Rimuovi rettangolo.
3.   Visualizza.
4.   Esci.
1
Vertice sud-ovest in formato (x,y): (0,0)
Vertice nord-est in formato (x,y): (1,1)
1.   Aggiungi rettangolo.
2.   Rimuovi rettangolo.
3.   Visualizza.
4.   Esci.
1
Vertice sud-ovest in formato (x,y): (-0.5,-0.5)
Vertice nord-est in formato (x,y): (1,2)
1.   Aggiungi rettangolo.
2.   Rimuovi rettangolo.
3.   Visualizza.
4.   Esci.
3
r1={(0,0), (1,1)}
r2={(-0.5,-0.5), (1,2)}
1.   Aggiungi rettangolo.
2.   Rimuovi rettangolo.
3.   Visualizza.
4.   Esci.
```

FIGURA 1. Esempio di esecuzione del programma, Esercizio 2.

Se l'utente sceglie 1, il programma chiede all'utente di inserire le coordinate del vertice sud-ovest e del vertice nord-est del nuovo rettangolo, e lo aggiunge alla lista usando la funzione `realloc` del file di intestazione `stdlib.h`. Se la riallocazione della memoria non riesce, il programma visualizza un messaggio d'errore e torna al menu.

Se l'utente sceglie 2, il programma elimina dalla lista di rettangoli l'ultimo rettangolo inserito dall'utente, e torna al menu. Se la riallocazione della memoria non riesce, il programma visualizza un messaggio d'errore e torna al menu. Se la lista è vuota nel momento in cui l'utente sceglie 2, il programma avvisa l'utente e torna al menu.

Se l'utente sceglie 3 il programma visualizza l'elenco di rettangoli correntemente memorizzati nella lista, e torna al menu. Se la lista è vuota, il programma avvisa l'utente, e torna al menu. Implementate una funzione ausiliaria

```
void visualizza(int n)
```

che riceva in ingresso la lunghezza della lista corrente, e visualizzi a video l'elenco dei rettangoli.

Si vedano le Figg. 1 e 2 per due esempi d'esecuzione.

ESERCIZIO 3

Calcolo del perimetro medio e del rettangolo di area massima.

Punti: 10.

Completate il menu implementato all'esercizio precedente nel modo seguente:

1. Aggiungi rettangolo.

```

Vincenzos-MacBook-Air:Soluzione enzo$ ./a.out
1.  Aggiungi rettangolo.
2.  Rimuovi rettangolo.
3.  Visualizza.
4.  Esci.
1
Vertice sud-ovest in formato (x,y): (0,0)
Vertice nord-est in formato (x,y): (1,1)
1.  Aggiungi rettangolo.
2.  Rimuovi rettangolo.
3.  Visualizza.
4.  Esci.
1
Vertice sud-ovest in formato (x,y): (-1,-1)
Vertice nord-est in formato (x,y): (0.1,0.2)
1.  Aggiungi rettangolo.
2.  Rimuovi rettangolo.
3.  Visualizza.
4.  Esci.
3
r1={(0,0), (1,1)}
r2={(-1,-1), (0.1,0.2)}
1.  Aggiungi rettangolo.
2.  Rimuovi rettangolo.
3.  Visualizza.
4.  Esci.
2
1.  Aggiungi rettangolo.
2.  Rimuovi rettangolo.
3.  Visualizza.
4.  Esci.
3
r1={(0,0), (1,1)}
1.  Aggiungi rettangolo.
2.  Rimuovi rettangolo.
3.  Visualizza.
4.  Esci.
2
1.  Aggiungi rettangolo.
2.  Rimuovi rettangolo.
3.  Visualizza.
4.  Esci.
3
Nessun rettangolo in memoria.

```

FIGURA 2. Esempio di esecuzione del programma, Esercizio 2.

2. Rimuovi rettangolo.
3. Visualizza.
4. Media perimetri.
5. Max area.
6. Esci.

Se l'utente digita 4, il programma calcola e visualizza il perimetro medio dell'elenco di rettangoli correntemente memorizzati. Se la lista è vuota, il programma avvisa l'utente e torna al menu. Per implementare questa voce del menu, scrivete la funzione ausiliaria di nome

`mediaper`

che calcola la media dei perimetri, della quale dovrete stabilire il prototipo appropriato.

Se l'utente digita 5, il programma visualizza il primo rettangolo dell'elenco correntemente memorizzato la cui area sia massima fra quelle dei rettangoli dell'elenco. Il programma visualizza anche il valore dell'area. Se la lista è vuota, il programma avvisa l'utente e torna al menu. Per implementare questa voce del menu, scrivete la funzione ausiliaria di nome

`areamax`

che restituisce un puntatore al rettangolo di area massima, della quale dovrete stabilire il prototipo appropriato.

```

Vincenzos-MacBook-Air:Soluzione enzo$ ./a.out
1.  Aggiungi rettangolo.
2.  Rimuovi rettangolo.
3.  Visualizza.
4.  Media perimetri
5.  Max area.
6.  Esci.
1
Vertice sud-ovest in formato (x,y): (0,0)
Vertice nord-est in formato (x,y): (1,1)
1.  Aggiungi rettangolo.
2.  Rimuovi rettangolo.
3.  Visualizza.
4.  Media perimetri
5.  Max area.
6.  Esci.
4
La media dei perimetri dei rettangoli e' 4.
1.  Aggiungi rettangolo.
2.  Rimuovi rettangolo.
3.  Visualizza.
4.  Media perimetri
5.  Max area.
6.  Esci.
1
Vertice sud-ovest in formato (x,y): (-1,-1)
Vertice nord-est in formato (x,y): (2,2)
1.  Aggiungi rettangolo.
2.  Rimuovi rettangolo.
3.  Visualizza.
4.  Media perimetri
5.  Max area.
6.  Esci.
4
La media dei perimetri dei rettangoli e' 8.
1.  Aggiungi rettangolo.
2.  Rimuovi rettangolo.
3.  Visualizza.
4.  Media perimetri
5.  Max area.
6.  Esci.

```

FIGURA 3. Esempio di esecuzione del programma, Esercizio 2.

Le altre voci del menu funzionano esattamente come nell'esercizio precedente. Si vedano le Figg. 3 e 4 per due esempi d'esecuzione.

Istruzioni per la consegna

- Funzioni da consegnare (in un unico file sorgente).

```

main
stampa
base
alt
area
per
visualizza
mediaper
areamax

```

Tutte le eventuali altre funzioni ausiliarie che abbiate scritto.

- Sito per la consegna:

<https://upload.mat.unimi.it>

```

Vincenzos-MacBook-Air:Soluzione enzo$ ./a.out
1.   Aggiungi rettangolo.
2.   Rimuovi rettangolo.
3.   Visualizza.
4.   Media perimetri
5.   Max area.
6.   Esci.
1
Vertice sud-ovest in formato (x,y): (0,0)
Vertice nord-est in formato (x,y): (1,1)
1.   Aggiungi rettangolo.
2.   Rimuovi rettangolo.
3.   Visualizza.
4.   Media perimetri
5.   Max area.
6.   Esci.
1
Vertice sud-ovest in formato (x,y): (-1,-1)
Vertice nord-est in formato (x,y): (0.1,0.1)
1.   Aggiungi rettangolo.
2.   Rimuovi rettangolo.
3.   Visualizza.
4.   Media perimetri
5.   Max area.
6.   Esci.
5
Il rettangolo di area massima e' r={{(-1,-1), (0.1,0.1)}}.
La usa area e' pari a 1.21.
1.   Aggiungi rettangolo.
2.   Rimuovi rettangolo.
3.   Visualizza.
4.   Media perimetri
5.   Max area.
6.   Esci.
2
1.   Aggiungi rettangolo.
2.   Rimuovi rettangolo.
3.   Visualizza.
4.   Media perimetri
5.   Max area.
6.   Esci.
5
Il rettangolo di area massima e' r={{(0,0), (1,1)}}.
La usa area e' pari a 1.

```

FIGURA 4. Esempio di esecuzione del programma, Esercizio 2.

- Autenticarsi con le proprie credenziali di posta elettronica d'ateneo.
- Consegnare solo il codice sorgente (file *.c), non l'eseguibile.

(V. Marra) DIPARTIMENTO DI MATEMATICA *Federigo Enriques*, UNIVERSITÀ DEGLI STUDI DI MILANO, VIA CESARE SALDINI, 50, I-20133 MILANO
E-mail address: vincenzo.marra@unimi.it