

Fondamenti di Informatica A

Ing. Elettronica e delle Telecomunicazioni

Esercitazione 5

14-15 Novembre 2006

Puntatori e funzioni

ESERCIZIO n° 1:

Realizzare un programma che, facendo uso di un vettore allocato dinamicamente, gestisca i dati relativi ai risultati di una gara di sci. In particolare, una volta noto il numero N di partecipanti alla gara, il programma dovrà allocare dinamicamente un vettore, nel quale ogni elemento rappresenta il risultato di un diverso atleta, tenendo conto che per ogni atleta è necessario memorizzare:

- **Numero:** un intero che rappresenta il numero di pettorale e individua univocamente lo sciatore;
- **Cognome:** una stringa che contiene il cognome dell'atleta;
- **Nome:** una stringa che contiene il nome dell'atleta;
- **Prima_manche:** un intero che rappresenta il numero di secondi impiegati per completare la prima manche della gara; (se questo valore è uguale a -1, significa che lo sciatore è caduto durante la manche, ed è stato quindi eliminato)
- **Seconda_manche:** un intero che rappresenta il numero di secondi impiegati per completare la seconda manche (se questo valore è uguale a -1, significa che lo sciatore è caduto durante la manche, ed è stato quindi eliminato).

Naturalmente, vince la gara lo sciatore che ha ottenuto il **tempo totale** (Prima_manche + Seconda_manche) **minimo** tra tutti.

Una volta acquisiti i dati relativi alla gara, il programma dovrà:

1. Stampare numero, nome e cognome degli sciatori eliminati;
2. Stampare numero, nome e cognome del vincitore. A tal scopo utilizzare una funzione con intestazione del tipo `struct atleta vincitore(struct atleta *v);`
3. Stampare la **graduatoria finale** degli sciatori non eliminati (cioè l'elenco ordinato in ordine di tempo totale crescente), visualizzando, per ogni sciatore numero, nome, cognome e tempo totale. Utilizzare una procedura con intestazione del tipo `void ordina(struct atleta *v);` per ordinare gli atleti.

ESERCIZIO n° 2:

Definire una funzione **sommatoria** che, dati due valori reali x e y, e un intero positivo N, calcoli come risultato la funzione reale:

$$\sum_{i=1}^N (x + y)^i$$

La funzione dovrà quindi prevedere un'intestazione del tipo:

```
float sommatoria(float x, float y, int N)
```

Realizzare un programma che, dati x, y e N da standard input, mediante la funzione **sommatoria** calcoli la sommatoria sui valori effettivamente letti, e successivamente la stampi.

ESERCIZIO n° 3:

Realizzare un programma in grado di effettuare alcune operazioni sul figure geometriche di tipo triangolari.

A questo scopo progettare un tipo di dato strutturato non primitivo triangolo in grado di rappresentare la figura geometrica mediante i tre lati e l'altezza. Il programma dovrà essere in grado di calcolare, per ogni triangolo dato da input, mediante opportune funzioni, l'area e il perimetro:

Utilizzare inoltre una procedura che calcoli e stampi le caratteristiche.

Le funzioni e la procedura dovranno avere un'intestazione del tipo:

```
float Area(struct triangolo T);
```

```
int Perimetro (struct triangolo T);
```

```
void Stampa (struct triangolo V[], int n);
```