

Esercizio 0

In questo esercizio usiamo funzioni. Si tratta di scrivere un programma che dichiara un array A di 100 interi, poi legge da cin un intero $n \leq 100$ e infine legger da cin n interi che inserirà nelle prime n posizioni dell'array A. La cosa nuova è che la lettura degli n valori nell'array A va eseguita da una apposita funzione.

La funzione si chiama leggi e, per poter eseguire il suo compito, deve ricevere l'array A e il valore di n. La dichiarazione di leggi è: `void leggi(int X[], int m)` ed essa verrà invocata dal main con il comando:

`leggi(A,n)`

Per il resto il corpo di leggi sarà semplicemente un ciclo di lettura del tutto simile a quelli già visti:

```
void leggi(int X[], int m) //void significa che la funziona non ritorna alcun risultato
{
int i=0;
while(i < m)
{
    cin >> X[i];
    i=i+1;
}
}
```

Per semplificare il seguito, assumiamo che gli n valori letti in A siano tutti distinti. Dopo aver letto gli n valori distinti in A, vogliamo calcolare la lunghezza della massima porzione crescente di A che inizia nella posizione 0 di A. Precisamente vogliamo calcolare il massimo $1 \leq k \leq n$ tale che $A[0..k-1]$ è tale che per ogni i in $[0..k-2]$, vale che $A[i] < A[i+1]$.

Esempio: se $n=10$ e $A=[2,3,4,0,8,9,10,11,12,15]$, il k che cerchiamo è $k=3$ e corrisponde alla porzione crescente $[2,3,4]$ di lunghezza 3. La porzione è massima visto che l'elemento successivo è 0 che è minore di 5.

Se $n=5$ e $A=[1,2,3,4,5]$, k sarebbe 5, cioè l'intero array $A[0..4]$. Infine se $n=5$ e $A=[5,1,2,3,4]$ ovviamente $k=1$ e la porzione sarebbe solo $A[0]=5$ di lunghezza 1.

E' possibile realizzare il calcolo della lunghezza k della porzione massima crescente (che inizia in 0) con una funzione. La funzione potrebbe chiamarsi trovaporz e avrebbe la seguente segnatura:

`int trovaporz(int Y[], int q)` dove i parametri Y e q servono a ricevere l'array A e il numero di valori n da considerare (come per la funzione leggi), mentre il k sarebbe il valore che la funzione restituisce attraverso un'apposito comando `return k`; da mettere alla fine della funzione stessa.

Durante il laboratorio, è normale e anche auspicato che gli studenti chiedano aiuto al docente per realizzare questa funzione.

Correttezza: Ogni funzione deve avere la propria PRE e POST.