Esercitazione 2

Array, cicli e stream

15-16-17 ottobre 2018

Corso di Laboratorio 1A, A.A. 18-19
Laurea triennale in Fisica, Università di Genova

Lo scopo di questa esercitazione è prendere dimestichezza con i cicli for e while, con gli array e con la lettura/scrittura su file.
I primi due esercizi (svolti) permettono di familiarizzare con la sintassi. Gli esercizi successivi usano i cicli e l'i/o per l'implementazione di algoritmi matematici (ricerca di massimo/minimo, massimo comun divisore e numeri primi).

- Svolgete gli esercizi secondo la traccia riportata nel resto di questo documento
- Seguite le istruzioni riportate qui per la consegna degli esercizi di laboratorio
- IMPORTANTE: la consegna deve essere effettuata da entrambi i componenti del gruppo

Esercizio 1

Calcolo della somma dei primi n numeri naturali

```
#include <iostream>
using namespace std;
int main(){
  int n;
  cout << "inserisci n" << endl;
  cin >> n;
  double sum=0;
  for (int i=1;i<=n;i++){
    sum = sum + i;
  }
  cout << "La somma vale " << sum << endl;
}
```

Questo esercizio ha lo scopo di familiarizzare con il ciclo for.
Verificare che, se nel ciclo viene eseguita una sola istruzione, le parentesi graffe possono essere omesse.

Materiale da consegnare: esercizio1.cpp (entro la fine dell'esercitazione)

Esercizio 2

Dichiarazione di un array, lettura delle componenti da terminale, stampa.

```
#include <iostream>
using namespace std;
int main(){
  int n=10;
  int v[n];
  for (int i=0;i<n;i++){
    cout << "Inserire v[" << i << "]: ";
    cin >> v[i];
  }
  for (int i=0;i<n;i++){
    cout << "La componente " << i  << " del vettore v vale "
         << v[i] << endl;
  }
  return 0;
}
```

Questo programma usa un ciclo for per leggere un array da terminale, e poi un altro ciclo for per stampare il contenuto dell'array su terminale.

Materiale da consegnare: esercizio2.cpp (entro la fine dell'esercitazione)

Esercizio 3

Creare un programma che legga da terminale un array di numeri reali a 10 componenti e:

- Determini e stampi su schermo la somma delle componenti
- Determini e stampi su schermo la componente con valore minimo e quella con valore massimo (si supponga che le componenti minima e massima siano uniche)

Suggerimenti

Sviluppare il programma a partire dall'esercizio 2, la cui struttura e' molto simile.

Algoritmo per il calcolo della somma

Si puo' usare un algoritmo analogo a quello dell'esercizio 1

Algoritmo per la ricerca di massimo e minimo

- Si crea un variabile temporanea per il minimo/massimo a cui si assegna la prima componente dell'array
- All'interno di un ciclo, si procede quindi a confrontare tale minimo/massimo con le altre componenti del vettore
- Se una componente è minore del minimo temporaneo o maggiore del massimo temporaneo, si aggiornano il minimo (o il massimo) a quel valore.

Materiale da consegnare: esercizio3.cpp (entro la fine dell'esercitazione)

Facoltativo

Modificare il programma dell'esercizio 3 in modo che legga l'array con i valori da un file vettori.dat (che dovrete creare) nel formato

N
x_1
x_2
...
x_N

in cui N nella prima riga indica la dimensione dell'array da leggere.

Materiale da consegnare: esercizio3bis.cpp (entro la fine dell'esercitazione)

Esercizio 4

Determinare il massimo comun divisore (MCD) tra due numeri (algoritmo di Euclide, http://en.wikipedia.org/wiki/Euclidean_algorithm)

Siano dati due numeri naturali a e b:

- Si controlla se b è zero. Se lo è, a è il MCD
- Se non lo è, si divide a / b e si assegna ad r il resto della divisione.
- Se r = 0 allora b è il MCD cercato.
- Altrimenti si assegna a = b e b = r e si ripete nuovamente la divisione

Materiale da consegnare: esercizio4.cpp (entro la fine dell'esercitazione)

Facoltativo

Modificare il programma per fare in modo che a e b siano letti da riga di comando anziché durante l'esecuzione del programma

Materiale da consegnare: esercizio4bis.cpp (entro la fine dell'esercitazione)

Esercizio per casa

Realizzare un programma che, inserito da terminale un numero naturale si determini se questo è primo.

Suggerimenti

- Si inizializza una variabile booleana a vero
- Si testano con un ciclo for tutti i possibili divisori
- Non appena si trova un divisore si esce dal ciclo (break)
- Se alla fine del loop la variabile booleana è vera il numero è primo, altrimenti non lo è

Ottimizzazione

Provare a velocizzare il calcolo sfruttando le seguenti proprietà:

- I numeri primi (a parte 2) sono dispari: basta cercare tra i divisori dispari
- Se si cercano i divisori di N basterà controllare i divisori fino a sqrt(N) (perche'?)

Materiale da consegnare: compito2.cpp (entro venerdi 26/10, tramite AulaWeb)

Estensione facoltativa

Determinare i primi M numeri primi (con M costante numerica definita nel programma).

Suggerimento: per ogni numero intero si applica l'algoritmo sviluppato al punto precedente, contando i numeri primi che via via si trovano, fermandosi quando se ne sono trovati M.

Provate a verificare la velocità dell'algoritmo eseguendo il programma con

time ./eseguibile

(il comando time esegue il programma passato come argomento e, alla fine, stampa il tempo di esecuzione)

Materiale da consegnare: compito2bis.cpp (entro venerdi 26/10, tramite AulaWeb)