

Esercizi per il laboratorio del 17/1/2014

Esercizio 1: Si tratta di aprire un file "input" ed un file "output" che saranno usati rispettivamente per leggere interi e per stampare il risultato. Controllare che l'apertura dei file abbia avuto successo e, in caso sia così, il programma deve leggere un intero $n \geq 0$ e dopo deve leggere altri n interi. Alla fine della lettura il programma deve stampare il valore iniziale di n , seguito dal massimo tra gli n valori letti.

La pre- e post-condizione del programma sono:

PRE=("input" contiene un intero $n \geq 0$, seguito da n interi)

POST=("output" deve contenere n e, se $n=0$, conterrà il valore INT_MIN, mentre se $n>0$, allora conterrà il massimo tra gli n interi che su "input" seguono (vedi PRE) il valore n)

Un programma che soddisfa queste PRE e POST segue:

```
#include<iostream>
#include<fstream>
#include<climits>
using namespace std;
main()
{
    int n, n_use, x, max=INT_MIN;
    ifstream INP("input");
    ofstream OUT("output");
    if(INP && OUT)
    {
        INP>>n;
        n_use=n;
        while(n_use>0)
        {
            INP>>x;
            if(x>max)
                max=x;
            n_use=n_use-1;
        }
        OUT<< n<< ' ' << max;
        INP.close();
        OUT.close();
    }
    else
        cout<<"errore nell'apertura dei file"<<endl;
}
```

NOTA: il programma usa INT_MIN per inizializzare la variabile max destinata a contenere il massimo dei valori letti.

Esercizio 2: Il problema è lo stesso dell'esercizio 1, ma ora non si vuole usare INT_MIN (né qualsiasi altro valore che non sia contenuto nel file "input") per inizializzare la variabile destinata a contenere il massimo.

La PRE e la POST del nuovo programma sono:

PRE = come per l'Esercizio 1,

POST=(se $n=0$, "output" deve contenere la stringa " $n=0$ ", se $n>0$, "output" deve contenere n seguito dal massimo degli n valori che seguono su "input" il valore n)

Esercizio 3: Come per i precedenti si vuole un programma capace di leggere dal file "input" un intero n , poi, ancora da "input", n interi e alla fine deve stampare su "output" il massimo seguito dal minimo dei numeri letti. Di questo esercizio si richiedono 3 versioni con diverse coppie di PRE e POST:

PRE1=("input" contiene $n \geq 0$, seguito da (almeno) n interi)

POST1=("output" contiene sempre n , che, a seconda del valore di n , è seguito dai seguenti valori:

i) se $n=0$, è seguito da INT_MIN e INT_MAX,

ii) se $n \geq 1$, è seguito dal massimo e dal minimo dei valori che seguono n su "input")

PRE2=PRE1

POST2=("output" contiene sempre n , che, a seconda del valore di n , è seguito dai seguenti valori:

i) se $n=0$, è seguito da INT_MIN e INT_MAX,

ii) se $n=1$, è seguito dall'unico valore che segue n su "input" e poi da INT_MAX,

iii) se $n \geq 2$, è seguito dal massimo e dal minimo dei valori che seguono n su "input")

PRE3=("input" contiene $n \geq 1$ seguito da n valori interi)

POST3=("output" contiene n seguito dal massimo e dal minimo degli n valori che seguono n su "input").

Esercizio 4: come per i precedenti esercizi, "input" contiene un intero n seguito da n interi. Il programma deve calcolare il secondo (più grande) tra gli n valori che legge. Come nell'Esercizio 3, il programma da fare dipende molto dalla pre-condizione che si assume. Considerate almeno queste 2 possibili pre-condizioni:

PRE1="input" contiene $n \geq 0$ seguito da n interi)

PRE2="input" contiene $n > 1$ seguito da n interi)

Con PRE1 si deve considerare anche i casi $n=0$ e $n=1$ in cui non esiste un secondo. Quindi, per questi casi, la post-condizione richiede che venga stampato qualche valore convenzionale (per esempio "non esiste un secondo", o altro). Invece se si assume PRE2, un secondo ci sarà sicuramente e quindi la corrispondente POST2 sarà più semplice della POST1.

Tenendo conto di queste osservazioni, scrivete una POST1 e POST2 che vadano d'accordo con PRE1 e PRE2, rispettivamente, e poi scrivete 2 programmi che siano corretti rispetto a queste 2 coppie di asserzioni.