Corso di Fondamenti di Informatica
Prof Aldo Franco Dragoni

prova di programmazione del 25 febbraio 2013

Avvertenze

- Consegnare solo fogli formato A4.
- In ordine di preferenza usare inchiostro nero, matita, inchiostro blu.
- In testa a ciascun foglio scrivere: cognome, nome, numero progressivo di pagina rispetto al totale; esempio per il secondo foglio di 3 consegnati: Giuseppe Russo 2/3
- Mantenere sul banco il libretto o altro documento di riconoscimento fino a controllo avvenuto
- Nient'altro deve trovarsi sul banco: non è consentito consultare libri, dispense, appunti, ecc.
- La correzione di riferimento per l'autovalutazione verrà fornita sul sito internet del Corso
- La consegna delle fotocopie dei compiti avverrà al termine della correzione
- Chi si presenta all'orale deve portare la propria soluzione, corretta ed autovalutata a penna rossa.

Un file di testo `NEVE.DAT` contiene lo stato delle piste di località sciistiche. Ciascuna riga del file contiene il *nome* della località (al più 13 caratteri senza spazi), uno *spazio bianco*, la *quantità minima* in centimetri della neve sulle piste di tale località (`float`), uno *spazio bianco* e la *quantità massima* (`float`).
Il candidato definisca una struttura chiamata `localita` in grado di contenere i dati di cui sopra e leggendo dal file costruisca un vettore di `localita` chiamato `array`. Costruisca poi un vettore chiamato `array1` ottenuto portando nella sua prima parte tutti gli elementi di `array` la cui quantità di neve sia superiore a 10 centimetri e inferiore a 30 centimetri e nella sua seconda parte i restanti elementi di `array`.
Successivamente stampi solo la prima parte di `array1`.
Infine, ripartendo dal file `NEVE.DAT` stampi l'elenco delle località la cui quantità di neve massima non sia inferiore al 10% dalla quantità massima assoluta.
Il numero di righe del file non è noto a priori, ma comunque non superiore a 40

ESEMPIO: se il file `NEVE.DAT` è

```
rivisondoli 15 25
monteprata 15 40
montepiselli 20 50
sassotetto 5 50
frontignano 40 110
montecopiolo 0 10
pratoselva 12 28
```

si ha:

```
Localita' con l'innevamento richiesto (>10 e <30 cm)
rivisondoli 15 25
pratoselva 12 28

Localita' innevate non inferiore al 10%
rivisondoli 15 25
monteprata 15 40
montepiselli 20 50
sassotetto 5 50
frontignano 40 110
pratoselva 12 28
```

60131 ANCONA - ITALY - Via Brecce Bianche - Segreteria phone +39(71)2204242
fax +39(71)2204835 - e-mail: segreteria.deit@univpm.it

```cpp
#include <iostream>
#include <stdlib.h>
#include <fstream>
#include <cstring>
#define MAXSTAZIONI 40

using namespace std;


typedef struct {
    char stazione[14];
    float min;
    float max;
} localita;

int Carica_da_File(localita ar[],char *nomefile);
int SelezionaStazioni(localita ar[],localita ar1[],int tot);
void Stampa(localita ar1[],int sel);
float CercaInnevamentoMassimo(localita ar[],int quante);
void StampaMigliori(char *nomefile,float max_livello);

int main(int argc, char *argv[])
{
      int quante;
    int selezionate;
    float max_innevamento;
    localita array[MAXSTAZIONI];
    localita array1[MAXSTAZIONI];
    char nomefile[10];
    strcpy(nomefile,"NEVE.DAT ");
    quante=Carica_da_File(array,nomefile);

    selezionate=SelezionaStazioni(array,array1,quante);
    if(selezionate>0)
        Stampa(array1,selezionate);
    max_innevamento=CercaInnevamentoMassimo(array,quante);
    StampaMigliori(nomefile,max_innevamento);

  system("PAUSE");
  return 0;
}

int Carica_da_File(localita ar[],char nomefile[]){
  fstream input;
  int i=0;

  input.open(nomefile,ios::in);
  while(input.good()){
      input >> ar[i].stazione>>ar[i].min>>ar[i].max;
      i++;
  }
  input.close();
```

```
53    return i;
54
55  }
56
57  int SelezionaStazioni(localita ar[],localita ar1[], int tot){
58      int cercate=0,i,altre;
59      for(i=0;i<tot;i++)
60          if(ar[i].min>10 && ar[i].max<30)
61              ar1[cercate++]=ar[i];
62      altre=cercate;
63      for(i=0;i<tot;i++)
64          if(ar[i].min<=10 || ar[i].max>=30)
65              ar1[altre++]=ar[i];
66      return cercate;
67  }
68
69  void Stampa(localita ar1[],int sel){
70      int i;
71      cout <<"\nLocalita' con l'innevamento richiesto (>10 e <30 cm
    )\n";
72      for(i=0;i<sel;i++)
73          cout<<ar1[i].stazione<<' '<<ar1[i].min<<' '<<ar1[i].max<<
    endl;
74  }
75
76  float CercaInnevamentoMassimo(localita ar[],int quante){
77      float max_inn=0;
78      int i;
79      for(i=0;i<quante;i++)
80          if(ar[i].max>=max_inn)
81              max_inn=ar[i].max;
82      return max_inn;
83  }
84
85  void StampaMigliori(char *nomefile,float max_livello){
86    fstream input;
87    localita app;
88
89    input.open(nomefile,ios::in);
90
91      cout <<"\nLocalita' innevate non inferiore al 10%\n";
92      while(input.good()){
93        input >> app.stazione>>app.min>>app.max;
94          if(app.max>=max_livello/10)
95              cout<<app.stazione<<' '<<app.min<<' '<<app.max<<endl;
96              }
97  input.close();
98
99  }
100
101
102
```