Programozási alapismeretek
beadandó feladat
Zenelejátszás: A, Y, MM

Készítette: Prorok Ernest
Neptun-azonosító: FILSPA
E-mail: ernest.prorok@gmail.com

Kurzuskód: IP-08PAEG/10
Gyakorlatvezető neve: Horváth Győző

2014. november 30.

Tartalom

Felhasználói dokumentáció

Feladat

A Budapest-Székesfehérvár vasútvonalon egy vonat kalauza minden állomáson feljegyezte, hogy hányan szálltak fel a vonatra, illetve hányan szálltak le. (Budapesten biztos nincs leszálló, Székesfehérváron biztos nincs felszálló, aki leszállt, az nem száll vissza.)

Sorszám	Feladat szövege
P	Add meg azt az állomást, ahol a legtöbbel csökkent az utasok száma a vonaton!
Y	Add meg, hogy hány állomáson szállt le a vonatról mindenki!
E	Add meg a leszállók átlagos számát!
EE	Ellenőrizd, hogy volt-e olyan állomás, ahol több ember szállt volna le, mint ahányan éppen a vonaton voltak!

Környezet

IBM PC, exe futtatására alkalmas operációs rendszer (pl. Windows XP). Nem igényel egeret.

Használat

A program indítása

A program a bead\bin\Debug\bead.exe néven található a tömörített állományban. A bead.exe fájl kiválasztásával indítható.

A program bemenete

A program az adatokat a billentyűzetről vagy fájlból olvassa be a következő sorrendben:

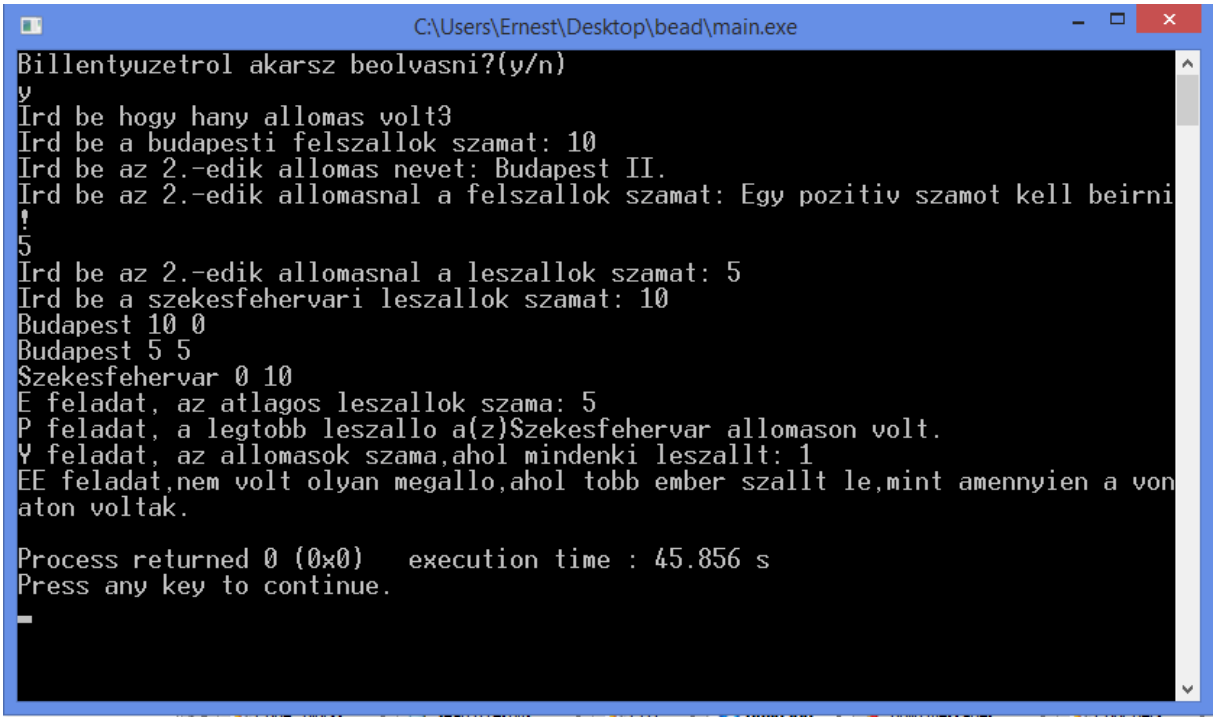
Sorindex	Adat	Magyarázat
0.	*bemód*	A beolvasás módja (fájlból olvasás esetén y, billentyűzetről olvasás esetén n)
1.	*N*	Az állomások száma.
2.	*állomás$_1$.megálló*	Az első megálló neve (egy szó).
3.	*állomás$_1$.felszállók*	Az első megállónál felszállók (egész).
4.	*állomás$_1$.leszállók*	Az első leszállók (egész).
...	...	
3*k–1.	*állomás$_k$.megálló*	A k-adik megálló neve (egy szó).
3*k.	*állomás$_k$.felszállók*	A k-adik megállón felszállók (egész).
3*k+1.	*állomás$_k$.leszállók*	A k-adik megállón leszállók (egész).

A program kimenete

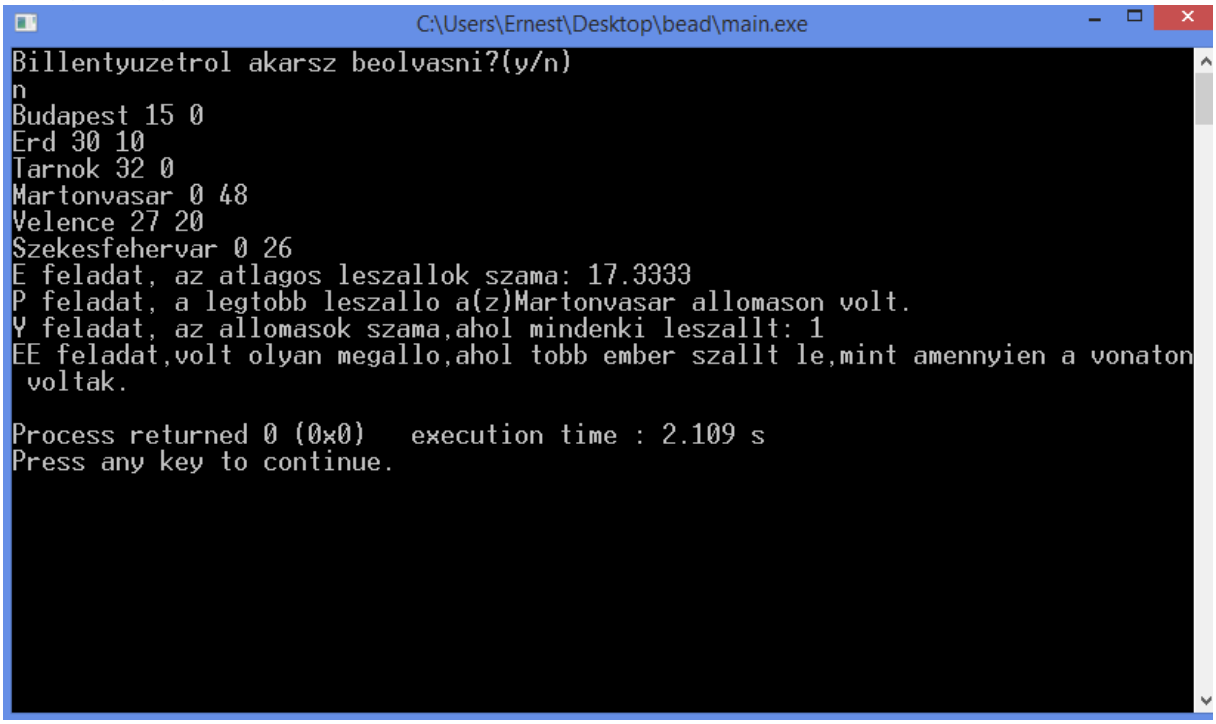
A program először kiírja a legtöbb utassal csökkenő megállót, majd a kiürült állomások számát, valamint a leszállók átlagos számát, végül pedig hogy volt-e olyan állomás, ahol több ember szállt volna le, mint ahányan éppen a vonaton voltak.

Minta bemenet és kimenet

Mintafutás billentyűzetről való beolvasás esetén:



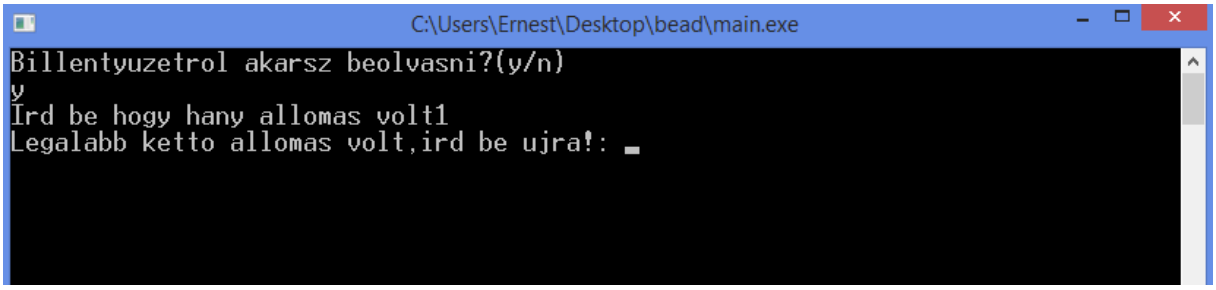
Mintafutás fájlból való beolvasás esetén:



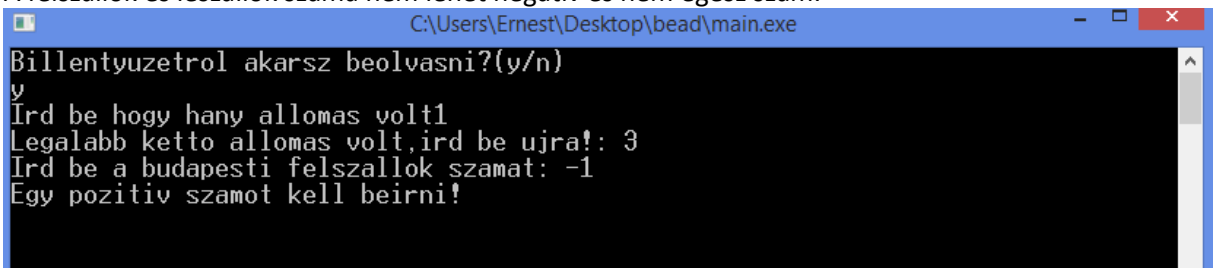
Hibalehetőségek:

Mintafutás hibás bemeneti adatok esetén:

A megállók számának 1-nél nagyobb természetes számnak kell lennie.



A felszállók és leszállók száma nem lehet negatív és nem egész szám.



Fejlesztői dokumentáció

Feladat

A Budapest-Székesfehérvár vasútvonalon egy vonat kalauza minden állomáson feljegyezte, hogy hányan szálltak fel a vonatra, illetve hányan szálltak le. (Budapesten biztos nincs leszálló, Székesfehérváron biztos nincs felszálló, aki leszállt, az nem száll vissza.)

Sorszám	Feladat szövege
P	Add meg azt az állomást, ahol a legtöbbel csökkent az utasok száma a vonaton!
Y	Add meg, hogy hány állomáson szállt le a vonatról mindenki!
E	Add meg a leszállók átlagos számát!
EE	Ellenőrizd, hogy volt-e olyan állomás, ahol több ember szállt volna le, mint ahányan éppen a vonaton voltak!

Specifikáció

Bemenet: $\mathrm{N} \in \mathbb{Z}, \mathrm{allomas} \in \mathrm{megallo}^{\mathrm{N}}, megallo * nev * felszallo * leszallo \in \mathrm{S}$
Előfeltétel: $\mathrm{N} > 2, \forall \mathrm{i}(1 \leq \mathrm{i} \leq \mathrm{N}): \mathrm{allomas_i}.\mathrm{nev} \neq '', \mathrm{allomas_i}.\mathrm{felszallo} \geq 0'', \mathrm{allomas_i}.\mathrm{leszallo} \geq 0$
P feladat
Kimenet: legtobbleszallo$\in \mathrm{S}$
Utófeltétel: megallo[i].leszallo>maximum
E feladat
Kimenet: atlagosleszallok$\in \mathbb{Z}$
Utófeltétel: atlag=atlag/db
Y feladat
Kimenet: utas$\in \mathbb{Z}$
Utófeltétel: utas=0
EE feladat
Kimenet: utas<0
Utófeltétel: l=false

Környezet

IBM PC, exe futtatására alkalmas operációs rendszer (pl. Windows XP). C++ fordítóprogram (gcc v4.4.1), Code::Blocks fejlesztői környezet.

Forráskód

A teljes fejlesztői anyag a bead nevű könyvtárban található meg. A fejlesztés során használt könyvtár-struktúra:

Állomány	Magyarázat
bead\bin\Debug\bead.exe	nyomkövethető állapotú futtatható kód
bead\obj\Debug\main.o	nyomkövethető állapotú, félig lefordított (object-) kód
bead\bead.cbp	projektfájl,
bead\main.cpp	C++ forrás
bead\bead.layout	layout file
bead\bin\Debug\teszt.txt	tesztfájl#1
bead\teszt.txt	tesztfájl#2
bead\bead.depend	depend file
bead\main.o	nyomkövethető állapotú, félig lefordított (object-) kód
bead\main.exe	nyomkövethető állapotú futtatható kód
bead/dokumentacio.pdf	a dokumentáció

Megoldás

Programértékek

Konstans

n:egész [az állomások max száma]

Típus

allomas=rekord(

nev: szöveg,

felszallo: egész,

leszallo: egész)

Változó

line tömb[megallo]

eldontes tömb[megallo]

szam egész

E double

EE logikai

Y egész

P tömb[megallo]

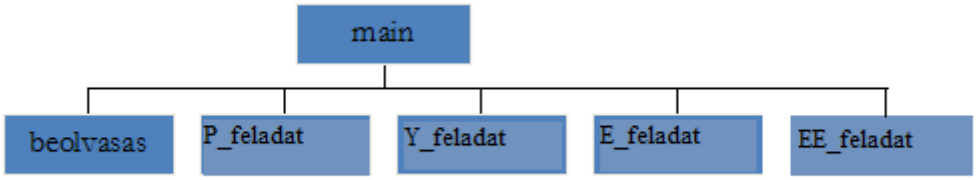
Programfelépítés

A program által használt modulok (és helyük):

bead – main.cpp

iostream, fstream, cstdlib, string, sstream – a C++ rendszer része.

Függvénystruktúra



Algoritmus

Az algoritmizálás szempontjából a részfeladatokat megoldó alprogramok érdekesek: Ezek algoritmusa a következő:

P feladat

```
nev:=BP
maximum:=0
        i=2..N
            i\ megallo[i].leszallo>maximum /h
            maximum:=megallo[i].leszallo
            nev:=megallo[i].nev            -
```

Y feladat

```
db:=0
utas:=megallo[1].felszallo
        i=2..N
            utas:=utas-megallo[i].leszallo
            i\ utas=0 /n
            db:=db+1                       -
            utas:=utas-megallo[i].felszallo
```

E feladat

```
db:=0
atlag:=0
        i=1..N
            db:=db+1
            atlag:=atlag+megallo[i].felszallo
atlag:=atlag/db
```

EE feladat

```
utas:=megallo[1].felszallo
l=false
        i=2..N
            utas=utas-megallo[i]leszallo
            i\              utas<0              /n
            l=true                      -
            utas:=utas+megallo.[i].felszallo
```

A kód

```
#include <iostream>
#include <fstream>
#include <string>
#include <cstdlib>
#include <sstream>
using namespace std;
int n=2;

struct allomas
{
   string nev;
   int leszallo;
   int felszallo;
};

double E (allomas a[]); //Deklaracio
string P (allomas a[]);
int Y (allomas a[]);
bool EE (allomas b[]);
void szam(int &a);

int main()
{
   string line;
  ifstream myfile ("teszt.txt");
  string eldontes;
  cout<<"Billentyuzetrol akarsz beolvasni?(y/n)"<<endl;
  cin>>eldontes;

  if(eldontes=="y")  // Billrol
  {
      cout<<"Ird be hogy hany allomas volt";
      szam(n);
      while(n<2)
      {   cout<<"Legalabb ketto allomas volt,ird be ujra!: ";
          szam(n);
      }
  }
  else //Fajl
  {
     getline(myfile,line);
     n=atoi(line.c_str());
  }
  allomas megallo[n];
  if(eldontes=="y")
  {


      megallo[0].nev="Budapest";
      megallo[0].leszallo=0;
      cout<<"Ird be a budapesti felszallok szamat: ";
```

```
      szam(megallo[0].felszallo);
      for(int i=1;i<(n-1);++i)
      {
         cout<<"Ird be az "<<i+1<<".-edik allomas nevet: ";
         cin>>megallo[i].nev;
                  cout<<"Ird be az "<<i+1<<".-edik allomasnal a felszallok szamat: ";
         szam(megallo[i].felszallo);
         cout<<"Ird be az "<<i+1<<".-edik allomasnal a leszallok szamat: ";
         szam(megallo[i].leszallo);
      }
      megallo[n-1].nev="Szekesfehervar";
      megallo[n-1].felszallo=0;
      cout<<"Ird be a szekesfehervari leszallok szamat: ";
      szam(megallo[n-1].leszallo);


   }
    else
    {
      for (int i=0;i<n;++i)
      {
         getline(myfile,line);
         stringstream ss(line);
         string buf;
         ss>>buf;
         megallo[i].nev=buf;
         ss>>buf;
         megallo[i].felszallo=atoi(buf.c_str());
         ss>>buf;
         megallo[i].leszallo=atoi(buf.c_str());
      }
      myfile.close();
    }

    for(int i=0;i<n;++i)
    {
      cout<<megallo[i].nev<<" "<<megallo[i].felszallo<<" "<<megallo[i].leszallo<<endl;
    }




    double atlagosleszallok = E(megallo);

cout << "E feladat, az atlagos leszallok szama: "<<atlagosleszallok<<endl;

string legtobbleszallo = P(megallo);

cout<<"P feladat, a legtobb leszallo a(z)"<<legtobbleszallo<<" allomason volt."<<endl;

int db = Y(megallo);
```

```
cout<<"Y feladat, az allomasok szama,ahol mindenki leszallt: "<<db<<endl;

bool l= EE(megallo);
if(l==true)
{
   cout<<"EE feladat,volt olyan megallo,ahol tobb ember szallt le,mint amennyien a vonaton
voltak."<<endl;
}
else
{
   cout<<"EE feladat,nem volt olyan megallo,ahol tobb ember szallt le,mint amennyien a vonaton
voltak."<<endl;
}

   return 0;
}

double E (allomas a[])
{
   int db=0;
   double atlag=0;
   for(int i=0;i<n;++i)
   {
      ++db;
      atlag=atlag+a[i].leszallo;
   }
   atlag=atlag/db;
   return atlag;
}

string P (allomas a[])
{
   string nev="Budapest";
   int maximum=0;
   for(int i=1;i<n;++i)
   {
      if(a[i].leszallo>maximum)
      {
         maximum=a[i].leszallo;
         nev=a[i].nev;
      }
   }
   return nev;
}

int Y (allomas a[])
{
   int db=0;
   int utas=a[0].felszallo;
   for(int i=1;i<n;++i)
   {
      utas=utas-a[i].leszallo;
```

```
      if(utas==0)
      {
         ++db;
      }
      utas=utas+a[i].felszallo;
   }
   return db;
}

bool EE (allomas b[])
{
   int utas=b[0].felszallo;
   bool l=false;
   for(int i=1;i<n;++i)
   {
      utas=utas-b[i].leszallo;
      if(utas<0)
      {
         l=true;
      }
      utas=utas+b[i].felszallo;
   }
   return l;
}

void szam(int &a)
{

        cin >> a;
        while(cin.fail() || a<0 )
        {
                cout << "Egy pozitiv szamot kell beirni!"<< endl;
                cin.clear();
                cin.ignore(100, '\n');
                cin >> a;
        }
}
```

Tesztelés

Érvényes teszteset

Bemenet	
N = 3	
1	10
2	Budapest II. 5 5
3	10
Kimenet	
E	5
P	Székesfehérvár
Y	1

EE	nem

Érvénytelen teszteset

Bemenet	
N = 0	
Kimenet	
E	-
P	-
Y	-
EE	-

Fejlesztési lehetőségek

Amennyiben a felhasználó szeretné saját maga is létrehozhatna közvetlenül a programon keresztül állományokat, melynek adatait saját magunk adjuk meg a programon keresztül, illetve lehetőség lenne generált (random) adatokkal való munkára.