

**Programozási alapismeretek
beadandó feladat
Zenelejátszás: A, Y, MM**

*Készítette: Gipsz Jakab
Neptun-azonosító: ABC123
E-mail: gipszjakab@seholse.hu*

*Kurzuskód: IP-PROGALAP2014/02
Gyakorlatvezető neve: Szlávi Péter*

2014. március 19.

Tartalomjegyzék

FELHASZNÁLÓI DOKUMENTÁCIÓ	4
<i>Feladat</i>	4
<i>Környezet</i>	4
<i>Használat</i>	4
A program indítása	4
A program bemenete	4
A program kimenete	5
Minta bemenet és kimenet.....	5
Hibalehetőségek.....	5
FEJLESZTŐI DOKUMENTÁCIÓ.....	6
<i>Feladat</i>	6
<i>Specifikáció</i>	6
A_feladat.....	6
Y_feladat.....	6
MM_feladat	6
<i>Környezet</i>	6
<i>Forráskód</i>	7
<i>Megoldás</i>	7
Programparaméterek.....	7
Konstans.....	7
Típus	7
Változó.....	7
Programfelépítés	7
Függvénystruktúra	8
Algoritmus	8
A kód.....	9
<i>Tesztelés</i>	12
Érvényes tesztesetek	12
1.teszteset: be1.txt.....	12
2.teszteset: be2.txt.....	12
3.teszteset: be3.txt.....	13
Érvénytelen tesztesetek.....	13
4.teszteset: be4.txt.....	13
5.teszteset: be5.txt.....	13
<i>Fejlesztési lehetőségek</i>	13

Felhasználói dokumentáció

Feladat

Összeállítottunk egy zenelejátszási listát, amelyben N zeneszám van. A listában a lejátszás sorrendjében szerepelnek az előadók neve (egy szó), a dal címe (egy szó) és a hosszúsága másodpercekben. (Ugyanaz az előadó, ugyanazzal a dallal csak egyszer szerepel.)

Sorszám	Feladat szövege
A	Milyen hosszú ideig tart a lista lejátszása?
Y	Melyik a leghosszabb dal?
MM	Melyik előadónak van a legtöbb száma a listán (az első ilyen adjuk meg)?

Környezet

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

Használat

A program indítása

A program a *GipszJakab\bin\Debug\zeneAYMM.exe* néven található a tömörített állományban. A *zeneAYMM.exe* fájl kiválasztásával indítható.

A program bemenete

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

Sorindex	Adat	Magyarázat
0.	<i>bemód</i>	A beolvasás módja (fájlból olvasás esetén 1, billentyűzetről olvasás esetén 2)
1.	N	A lejátszási lista hossza.
2.	<i>zenék₁.előadó</i>	Az első szám előadójának neve (egy szó).
3.	<i>zenék₁.cím</i>	Az első szám címe (egy szó).
4.	<i>zenék₁.hossz</i>	Az első szám hossza másodpercben (egész).
...	...	
$3*k-1$.	<i>zenék_k.előadó</i>	A k -adik szám előadójának neve (egy szó).
$3*k$.	<i>zenék_k.cím</i>	A k -adik szám címe (egy szó).
$3*k+1$.	<i>zenék_k.hossz</i>	A k -adik szám hossza másodpercben (egész).

A program kimenete

A program először kiírja a lejátszási lista hosszát, majd a leghosszabb dal adatait, végül pedig annak az előadónak a nevét, akinek a legtöbb száma szerepel a listában.

Minta bemenet és kimenet

Minta futás billentyűzetről való beolvasás esetén:

```
zeneAYMM
ADATOK BEOLVASASA.
Adja meg a beolvasas modjat! (fajlbol - 1, billentyuzetrol - 2)
2
Hany szambol all a lejatszasi lista?
3
1. szam eloadoja:
Omega
1. szam cime:
Babylon
1. szam hossza:
246
2. szam eloadoja:
Katy_Perry
2. szam cime:
Firework
2. szam hossza:
234
3. szam eloadoja:
Adele
3. szam cime:
Someone_like_you
3. szam hossza:
285
A lejatszasi lista hossza: 765 mp.
A leghosszabb dal Adele: Someone_like_you, a hossza 285 mp.
A legtöbb szama a listaban a(z) Omega eloadonak van.

Process returned 0 (0x0)   execution time : 71.220 s
Press ENTER to continue.
```

Minta futás fájlból való beolvasás esetén:

A bemeneti állomány nevét a kiterjesztésével együtt kell megadni, és a bemeneti állományt közvetlenül a futtatható állomány mellé kell elhelyezni.

```
zeneAYMM
ADATOK BEOLVASASA.
Adja meg a beolvasas modjat! (fajlbol - 1, billentyuzetrol - 2)
1
Az allomany neve: be1.txt
A lejatszasi lista hossza: 765 mp.
A leghosszabb dal Adele: Someone_like_you, a hossza 285 mp.
A legtöbb szama a listaban a(z) Omega eloadonak van.

Process returned 0 (0x0)   execution time : 7.156 s
Press ENTER to continue.
```

Hibalehetőségek

A lejátszási lista hosszának 0-nál nagyobb egész számot kell megadni. A szám előadójának neve és a szám címe nem tartalmazhat szóközt, ehelyett célszerű az „_” karakter használata. A zeneszámok hossza szintén 0-nál nagyobb egész szám kell, hogy legyen.

Minta futás hibás bemeneti adatok esetén:

```
zeneAYMM
Adja meg a beolvasas modjat! (fajlbol - 1, billentyuzetrol - 2)
0
Hibas adat!
Adja meg a beolvasas modjat! (fajlbol - 1, billentyuzetrol - 2)
2
Hany szambol all a lejatszasi lista?
0
Hibas adat!
Hany szambol all a lejatszasi lista?
1
1. szam eloadoja:
Katy Perry
1. szam cime:
1. szam hossza:
0
Hibas adat!
1. szam hossza:
20
A lejatszasi lista hossza: 20 mp.
A leghosszabb dal Katy: Perry, a hossza 20 mp.
A legtöbb szama a listaban a(z) Katy eloadonak van.

Process returned 0 (0x0)   execution time : 24.202 s
Press ENTER to continue.
█
```

```
zeneAYMM
ADATOK BEOLVASASA.
Adja meg a beolvasas modjat! (fajlbol - 1, billentyuzetrol - 2)
1
Az allomany neve: be1
Fajl nyitasi hiba!
Az allomany neve: be1.txt
A lejatszasi lista hossza: 765 mp.
A leghosszabb dal Adele: Someone_like_you, a hossza 285 mp.
A legtöbb szama a listaban a(z) Omega eloadonak van.

Process returned 0 (0x0)   execution time : 15.211 s
Press ENTER to continue.
█
```

Fejlesztői dokumentáció

Feladat

Összeállítottunk egy zenelejátszási listát, amelyben N zeneszám van. A listában a lejátszás sorrendjében szerepelnek az előadók neve (egy szó), a dal címe (egy szó) és a hosszúsága másodpercekben. (Ugyanaz az előadó, ugyanazzal a dallal csak egyszer szerepel.)

Sorszám	Feladat szövege
A	Milyen hosszú ideig tart a lista lejátszása?
Y	Melyik a leghosszabb dal?
MM	Melyik előadónak van a legtöbb száma a listán (az első ilyet adjuk meg)?

Specifikáció

Bemenet: $N \in \mathbb{Z}$, $zenek \in Tzene^N$, $Tzene = eloado \times cim \times hossz$, $eloado, cim \in S$, $hossz \in \mathbb{Z}$

Előfeltétel: $N > 0, \forall i \ 1 \leq i \leq N : zenek_i.eloado \neq "", zenek_i.cim \neq "", zenek_i.hossz > 0$

A_feladat

Kimenet: $hossz \in \mathbb{Z}$

Utófeltétel: $hossz = \sum_{i=1}^N zenek_i.hossz$

Y_feladat

Kimenet: $max \in \mathbb{Z}$

Utófeltétel: $1 \leq max \leq N \wedge \forall i \ 1 \leq i \leq N : zenek_{max}.hossz \geq zenek_i.hossz$

MM_feladat

Kimenet: $maxea \in S$

Utófeltétel: $maxea \in \{ zenek_i.eloado : 1 \leq i \leq N \}$

$$\forall j \ 1 \leq j \leq N : \sum_{i=1}^N \mathbb{1}_{zenek_i.eloado=maxea} \geq \sum_{i=1}^N \mathbb{1}_{zenek_i.eloado=zenek_j.eloado}$$

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 GipszJakab nevű könyvtárban található meg. A fejlesztés során használt könyvtár-struktúra:

Allomány	Magyarázat
zeneAYMM\bin\Debug\zeneAYMM.exe	nyomkövethető állapotú futtatható kód
zeneAYMM\obj\Debug\main.o	nyomkövethető állapotú, félig lefordított (object-) kód
zeneAYMM\zeneAYMM.cbp	projekt fájl,
zeneAYMM\main.cpp	C++ forrás
zeneAYMM\zeneAYMM.layout	layout file
Teszt\teszt1.txt	teszt fájl#1
Teszt\teszt2.txt	teszt fájl#2
Teszt\teszt3.txt	teszt fájl#3
Teszt\teszt4.txt	teszt fájl#4

Megoldás

Programparaméterek

Konstans

MAXN : Egész(1000) [a zeneszámok maximális száma]

Típus

Tzene = Rekord(

 eloado: Szöveg,
 cím: Szöveg,
 hossz: Egész)

Változó

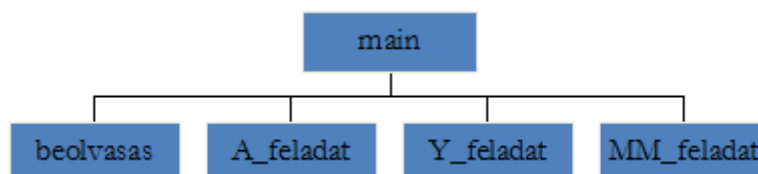
N: Egész
zenek: Tömb[1..MAXN : Tzene]
hossz: Egész
max: Egész
maxea: Szöveg

Programfelépítés

A program által használt modulok (és helyük):

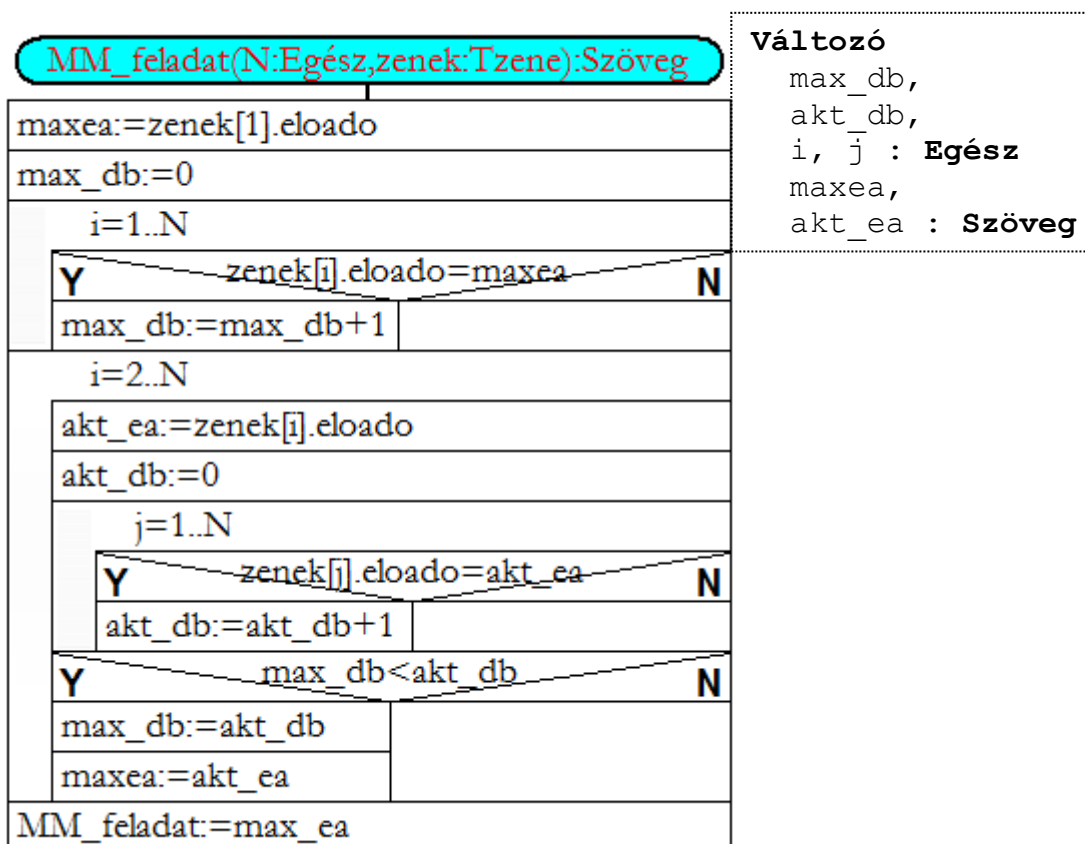
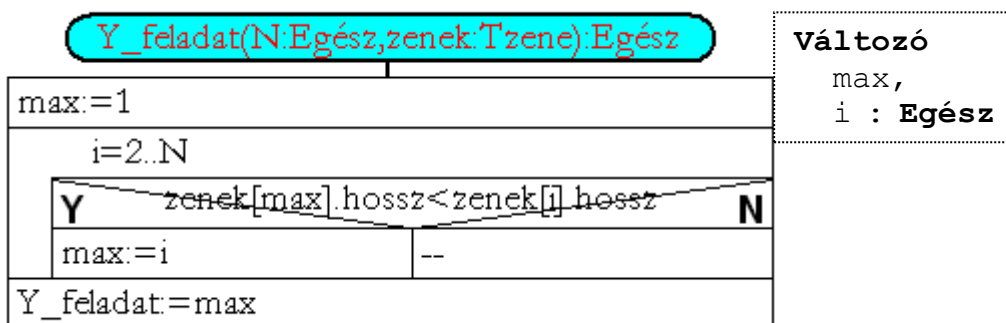
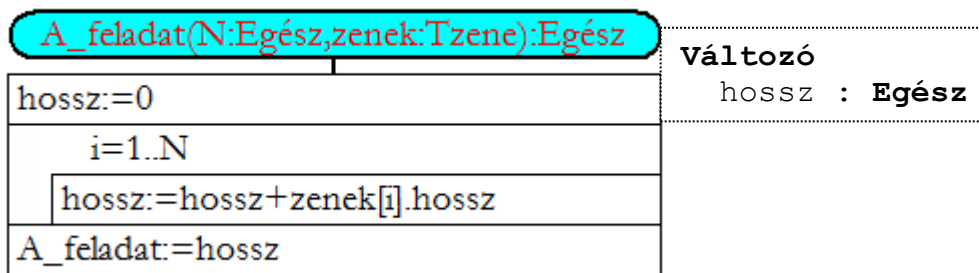
zeneAYMM – main.cpp
iostream, fstream, cstdlib – a C++ rendszer része.

Függvénystruktúra



Algoritmus

Az algoritmizálás szempontjából a részfeladatokat megoldó alprogramok érdekesek: Ezek algoritmusai a következők:



A kód

```
/*
Készítette: Gipsz Jakab
Neptun: ABC123
E-mail: gipszjakab@seholse.hu
Projekt címe: zeneAYMM
Feladat: Zenelejátás
        A, Y, MM
*/

#include <iostream>
#include <fstream>
#include <cstdlib>
using namespace std;
const int MAXN = 1000;

struct Tzene
{
    string eloado;
    string cim;
    int hossz;
};

void beolvasasC(int& N, Tzene zenek[MAXN]);
void beolvasasF(int& N, Tzene zenek[MAXN]);
int A_feladat(int N, Tzene zenek[MAXN]); //lejatszasi lista hossza
int Y_feladat(int N, Tzene zenek[MAXN]); //leghosszabb dal sorszama
string MM_feladat(int N, Tzene zenek[MAXN]); //leggyakoribb előadó

int main()
{
    int N;
    Tzene zenek[MAXN];
    bool hiba;
    cout << "ADATOK BEOLVASASA." << endl;
    int bemod;
    do {
        cout << "Adj meg a beolvasas modjat! (fajlbol - 1, billentyuzetrol - 2)" << endl;
        cin >> bemod;
        hiba = cin.fail() || (bemod!=1) && (bemod!=2);
        if (hiba){
            cout << "Hibas adat!" << endl;
            cin.clear(); string tmp; getline(cin, tmp, '\n');
        }
    }while(hiba);
    if (bemod==1){
        beolvasasF(N,zenek);
    }else{
        beolvasasC(N,zenek);
    }
    cout << "A lejatszasi lista hossza: " << A_feladat(N,zenek) << " mp." << endl;
    int max = Y_feladat(N,zenek);
    cout << "A leghosszabb dal " << zenek[max-1].eloado << ": " << zenek[max-1].cim
        << ", a hossza " << zenek[max-1].hossz << " mp." << endl;
    cout << "A legtobb szama a listaban a(z) " << MM_feladat(N, zenek)
        << " eloadonak van." << endl;
    return 0;
}

void beolvasasC(int& N, Tzene zenek[MAXN]){
    bool hiba;
    do{
        cout << "Hany szambol all a lejatszasi lista?" << endl;
        cin >> N;
        hiba = cin.fail() || N<1;
        if (hiba){
            cout << "Hibas adat!" << endl;
            cin.clear(); string tmp; getline(cin, tmp, '\n');
        }
    }
}
```

```

}while(hiba);
for(int i=1; i<=N; i++){
    do{
        cout << i << ". szam eloadoja: " << endl;
        cin >> zenek[i-1].eloado;
        hiba = cin.fail() || zenek[i-1].eloado=="";
        if (hiba){
            cout << "Hibas adat!" << endl;
            cin.clear(); string tmp; getline(cin, tmp, '\n');
        }
    }while(hiba);
    do{
        cout << i << ". szam cime: " << endl;
        cin >> zenek[i-1].cim;
        hiba = cin.fail() || zenek[i-1].cim=="";
        if (hiba){
            cout << "Hibas adat!" << endl;
            cin.clear(); string tmp; getline(cin, tmp, '\n');
        }
    }while(hiba);
    do{
        cout << i << ". szam hossza: " << endl;
        cin >> zenek[i-1].hossz;
        hiba = cin.fail() || zenek[i-1].hossz<1;
        if (hiba){
            cout << "Hibas adat!" << endl;
            cin.clear(); string tmp; getline(cin, tmp, '\n');
        }
    }while(hiba);
}
}

void beolvasasF(int& N, Tzene zenek[MAXN]){
    ifstream f;
    string fnev;
    bool hiba;
    do{
        cout << "Az allomany neve: "; cin >> fnev;
        f.open(fnev.c_str());
        if(hiba = f.fail()){
            cout << "Fajl nyitasi hiba!" << endl;
            f.clear();
        }
    }while(hiba);
    f >> N;
    for(int i=1; i<=N; i++){
        f >> zenek[i-1].eloado;
        f >> zenek[i-1].cim;
        f >> zenek[i-1].hossz;
    }
    f.close();
}

int A_feladat(int N, Tzene zenek[MAXN]){
    int hossz = 0;
    for (int i=1; i<=N; i++){
        hossz = hossz + zenek[i-1].hossz;
    }
    return hossz;
}

int Y_feladat(int N, Tzene zenek[MAXN]){
    int max = 1;
    for (int i=2; i<=N; i++){
        if (zenek[max-1].hossz < zenek[i-1].hossz){
            max = i;
        }
    }
    return max;
}

```

```

}
string MM_feladat(int N, Tzene zenek[MAXN]){
    string maxea = zenek[0].eloado;
    int max_db = 0;
    for (int i=1; i<=N; i++){
        if (zenek[i-1].eloado ==maxea){
            max_db ++;
        }
    }
    for (int i=2; i<=N; i++){
        string akt_ea = zenek[i-1].eloado;
        int akt_db = 0;
        for (int j=1; j<=N;j++){
            if (zenek[j-1].eloado ==akt_ea){
                akt_db ++;
            }
        }
        if (akt_db>max_db){
            max_db = akt_db;
            maxea = akt_ea;
        }
    }
    return maxea;
}
}

```

Tesztelés

Érvényes tesztesetek

1. *teszteset: be1.txt*

Bemenet	
N = 3	
1	Omega Babylon 246
2	Katy_Perry Firework 234
3	Adele Someone_like_you 285
Kimenet	
A	765
Y	3 (Adele: Someone_like_you , 285)
MM	Omega

2. *teszteset: be2.txt*

Bemenet	
N = 3	
1	Katy_Perry Firework 234
2	Adele Rolling_in_the_deep 246
3	Adele Someone_like_you 285
Kimenet	
A	765
Y	3 (Adele: Someone_like_you , 285)
MM	Adele

3. *teszteset: be3.txt*

Bemenet	
N = 10	
1	Omega Babylon 246
2	Katy_Perry Firework 234

3	Madonna Frozen 302
4	Madonna Vogue 294
5	Madonna Give_it_2_me 254
6	Madonna Hung_up 327
7	Madonna 4_minutes 240
8	Adele Rolling_in_the_deep 246
9	Adele Someone_like_you 285
10	Michael_Jackson Thriller 553
Kimenet	
A	2981
Y	10 (Michael_Jackson: Thriller, 553)
MM	Madonna

4. *teszteset: be4.txt*

Bemenet	
N = 10	
1	Omega Babylon 246
Kimenet	
A	246
Y	1 (Omega: Babylon, 246)
MM	Omega

Érvénytelen tesztesetek

5. *teszteset: be5.txt*

Nincs szám a lejátszási listában.

Bemenet	
N = 0	

Kimenet	
A	-
Y	-
MM	-

6. *teszteset: be6.txt*

Az előadók neve, ill. a szám címe szóközt tartalmaz. Ekkor a program nem tudja helyesen eltárolni az adatokat, ezért mind a három részfeladatra hibás eredményt kapunk.

Bemenet	
N = 2	
1	Katy Perry Firework 234
2	Adele Theres a fire 246
Kimenet	
A	-
Y	-
MM	-

Fejlesztési lehetőségek

1. Hibás bemenetek megkülönböztetése, a hibának megfelelő különböző hibüzenetek megadása.
2. Annak megoldása, hogy a zeneszámok előadója és címe tartalmazhasson szóközöket is.