Laboratorul 9

1. Exemplu de programare.

Problema

Să presupunem că lucraţi la o firma de avocatură care are de rezolvat un caz de discriminare sexuală. Firma a obţinut un fişier cu date care conţine salariile angajaţilor unei companii. Ca un prim pas în analiza datelor, vi se cere să calculaţi media veniturilor pentru femei şi pentru bărbaţi.

Intrări

Fişierul de intrare accesat prin stream-ul `fVenituri` conţine veniturile sub forma de numere reale, iar pe fiecare linie se găseşte un singur venit. Fiecare dată este precedată de un caracter (`'F'` pentru femeie sau `'B'` pentru bărbat). Acest cod este primul caracter de pe linie, urmat de un spaţiu si apoi de venitul corespunzător.

Ieşiri

Toate datele de intrare vor fi tipărite în ecou.
Numărul femeilor şi media veniturilor lor.
Numărul bărbaţilor şi media veniturilor lor.

Discuţie

Problema poate fi împărtiţă în doi paşi importanţi. Mai întâi se procesează datele de intrare, adică se contorizează şi se însumează veniturile pentru fiecare sex. Apoi se calculează mediile şi se afişează rezultatele. Cel de-al doilea pas este mai complicat şi presupune proiectarea unei bucle care cuprinde câteva subprobleme. Vom folosi un cadru general de proiectare a buclelor, parcurgând şapte paşi.

1. *Care este conditia de încheiere a buclei?* Condiţia de terminare este `EOF` în fişierul `fVenituri`. În pseudocod, putem scrie:

`WHILE NOT EOF` în `fVenituri`

2. *Cum trebuie iniţializate variabilele din condiţie?* Trebuie să deschidem fişierul de date şi să facem prima citire.

3. *Cum actualizăm variabilele din condiţie?* Trebuie să citim o nouă linie de date. Algoritmul rezultat până acum este:

Deschide `fVenituri` pentru citire (verifică şi corectitudinea operaţiei)
Citeşte sex şi venit din `fVenituri`
`WHILE NOT EOF` în `fVenituri`
.
.
. (Se repetă procesul)
Citeşte sex şi venituri din `fVenituri`

4. *Care este procesul care se repetă?* Pentru a calcula media trebuie să contorizăm numărul de femei şi să împărţim suma veniturilor lor la acest număr. La fel procedam şi pentru bărbaţi. Pentru că trebuie să facem o astfel de separare, procesul constă din patru paşi:

- contorizăm numărul de femei
- însumăm veniturile lor
- contorizăm numărul de bărbaţi

- însumăm veniturile lor.

5. *Cum trebuie iniţializat procesul?* `contorF` şi `sumaF` trebuie iniţializate cu 0. `contorB` şi `sumaB` trebuie iniţializate de asemenea cu 0.

6. *Cum se actualizează procesul?* Când citim venitul unei femei, `contorF` se incrementează şi venitul său este adunat la `sumaF`. Atunci când citim venitul unui bărbat, incrementăm `contorB` şi adunăm venitul lui la `sumaB`.

7. *Care este starea programului la ieşirea din buclă?* Streamul fişierului, `fVenituri` va fi în *fail state*; `contorF` va conţine numărul de valori de intrare precedate de `'F'`; `sumaF` va conţine suma veniturilor pentru valorile precedate de `'F'`; `contorB` va conţine numărul de valori care nu sunt precedate de `'F'`; `sumaB` va conţine suma valorilor care nu sunt precedate de `'F'`.

Din această descriere putem vedea că bucla va conţine o structură IF-THEN-ELSE cu o ramură pentru femei şi alta pentru bărbaţi. Fiecare ramură trebuie să incrementeze corect contorul de evenimente şi să adune valoarea citită la variabila corectă. Dupa încheierea buclei, vom avea suficiente informaţii pentru a calcula mediile şi a afişa datele cerute.

Ipoteză

Trebuie să presupunem că există cel puţin o linie de date precedată de `'F'` şi una precedată de `'B'` (sau altă literă). În caz contrar, programul generează erori datorate împărţirilor la 0. Pentru a corecta această eroare, ar trebui să inserăm câteva instrucţiuni `if` suplimentare care testează condiţiile generatoare de erori.

Programul

```
//*********************************************************
//Venituri.cpp
//Acest program citeste un fisier de venituri clasificate
//dupa sezul persoanelor care le realizeaza. Calculeaza
//media veniturilor pentru fiecare sex.
//*********************************************************
#include <iostream>
#include <iomanip>
#include <fstream>
using namespace std;

int main()
{
     char sex;
     int contorF;
     int contorB;
     double venit;
     double sumaF;
     double sumaB;
     double mediaF;
     double mediaB;
     ifstream fVenituri;

     cout.setf(ios::fixed, ios::floatfield);
```

```
     cout.setf(ios::showpoint);
     cout << setprecision(2);

     //Initializeaza conditia de oprire
     fVenituri.open("venituri.dat");
     if(!fVenituri)
     {
          cout << "Eroare: Nu se poate deschide fisierul de
venituri" << endl;
          return 1;
     }
     fVenituri >> sex >> venit;//Prima citire din fisier

     //Initializarea procesului
     contorF = 0;
     sumaF = 0.0;
     contorB = 0;
     sumaB = 0.0;

     while(fVenituri)
     {
          //Actualizarea procesului
          cout << "Sex: " << sex << " Venit: " << venit
               << endl;
          if(sex == 'F')
          {
               contorF++;
               sumaF = sumaF + venit;
          }
          else
          {
               contorB++;
               sumaB = sumaB + venit;
          }

          //Actualizarea conditiei de oprire
          fVenituri >> sex >> venit;
     }

     //Calculeaza mediile veniturilor
     mediaF = sumaF / static_cast<double>(contorF);
     mediaB = sumaB / static_cast<double>(contorB);

     //Afiseaza rezultatele
     cout << "Pentru " << contorF << " femei, "
          << "media veniturilor este" << mediaF << endl;
     cout << "Pentru " << contorB << " barbati, "
          << "media veniturilor este" << mediaB << endl;

     return 0;
}
```

Sugestii pentru test şi debugging

- Testaţi cu atenţie fiecare secţiune a programului pe care îl scrieţi.
- Fiţi atenţi la buclele infinite, adică atunci când condiţiile din instrucţiunea `while` nu devin niciodată false. Acesta este un motiv pentru care programul nu se mai opreşte. Dacă aţi creat bucle infinite, verificaţi logica şi sintaxa buclelor. Asiguraţi-vă ca nu aţi pus ; imediat după paranteza rotundă închisă de după condiţie:

```
while (condiţie);
      Instrucţiune
```

  Într-o buclă controlată de contor, fiţi siguri că variabila de control este incrementată în interiorul buclei. În buclele controlate de flag, verificaţi dacă flag-ul şi-a modificat valoarea. Fiţi atenţi la folosirea operatorilor = şi == în condiţiile din `while`.

```
while (someVar = 5) //Ar fi trebuit sa fie ==
```

  produce o buclă infinită. Valoarea expresiei este întotdeauna 5 şi este interpretată ca `true`.
- Verificaţi cu atenţie condiţia de terminare a buclei şi fiţi siguri că această condiţie poate apărea în timpul execuţiei buclei.
- Folosiţi funcţia `get` în locul operatorului de extracţie în buclele controlate de apariţia caracterului `newline`.
- Simulaţi execuţia buclelor pe hârtie. Studiaţi câţiva paşi de la începutul şi câţiva paşi de la finalul execuţiei buclei pentru a vă da seama cum funcţionează ea de fapt.
- Folosiţi un *debugger* pentru a studia evoluţia “cu încetinitorul” a programului.
- Dacă nu reuşiţi să vă daţi seama care sunt valorile pe care le ia o variabilă, folosiţi instrucţiuni de afişare utile la depanare:

```
cout << "beta = " << beta << endl;
```

2. Ce tipăreşte bucla următoare?

```
int number = 1;
while (number < 11)
{
     number++;
     cout << number << endl;
}
```

3. Rearanjând ordinea instrucţiunilor, fără a schimba modul în care sunt scrise, faceţi ca bucla de la exerciţiul anterior să tipărească numerele de la 1 la 10.

4. Câte iteraţii se realizează prin executarea codului de mai jos:

```
int number = 2;
bool done = false;
while (!done)
{
   number = number * 2;
   if (number > 64)
      done = true;
}
```

5. Scrieti un program pentru a verifica ce afişează următoarea structură de bucle imbricate:

```
int i=4;
while (i >= 1)
{
   int j = 2;
   while(j >= 1)
   {
      cout << j << ' ';
      j--;
   }
   cout << i << endl;
   i--;
}
```

6. Următorul fragment de cod ar trebui să afişeze numerele pare dintre 1 şi 15 (`n` este de tip `int`). El are, însă, două greşeli.

```
int n =2;
while (n != 15)
{
   n = n + 2;
   cout << n << ' ';
}
```

a) Care este ieşirea codului în varianta în care a fost scris?
b) Corectaţi codul pentru a funcţiona corect.

7. Arătaţi care este ieşirea următorului fragment de cod dacă toate variabilele sunt de tip `int`:

```
i = 1;
while (i <= 5)
{
   sum = 0;
   j = 1;
   while (j <= i)
   {
      sum = sum + 1;
      j++;
   }
   cout << sum << ' ';
   i++;
}
```

8. Scrieţi un program care contorizează câte numere 28 apar într-un fişier care conţine 100 de întregi.

9. Scrieţi un program care să tipărească următorul rezultat:

```
1
1 2
1 2 3
1 2 3 4
```

10. Scrieţi un program care citeşte dintr-un fişier numere întregi, contorizează şi tipăreşte separat numărul numerelor negative şi numărul numerelor pozitive. Dacă

apare valoarea 0, ea este ignorată. Programul se încheie la apariţia sfârşitului de fişier.

11. Scrieţi un program care tipăreşte secvenţa tuturor combinaţiilor de ore şi minute dintr-un interval de 60 minute pe 6 coloane şi 10 linii. Folosiţi formatul HH:MM (ex: 1:00, 12:59).

12. Modificaţi programul Venituri.cpp (punctul 1, Exemplu de programare) astfel încât:
a) Să tipărească un mesaj de eroare atunci când se citesc valori negative. Aceste valori vor fi omise din calcule;
b) Să nu facă afişări eronate atunci când fişierul de date nu conţine înregistrări pentru bărbaţi sau pentru femei;
c) Să nu ia în considerare liniile de date care nu sunt precedate de ‘F’ sau ‘B’.

13. Scrieţi programul C++ care acceptă ca intrări un numar întreg şi un caracter. Ieşirea va fi un romb desenat cu acel caracter, care îşi mareste diagonala până la dimensiunea dată de numărul întreg. În cazul în care numărul este par, el va fi incrementat până la următorul număr impar. De ex., dacă am introdus numărul 7 şi caracterul *, rombul va arăta astfel:

```
   *
  ***
 *****
*******
 *****
  ***
   *
```

14. Scrieţi programul C++ care acceptă la intrare un întreg mai mare decât 1 şi care calculează suma pătratelor numerelor de la 1 până la acel întreg. De ex., dacă numărul introdus este 4, programul calculează suma 1+4+9+16. Ieşirea va fi intregul introdus si suma, însoţite de mesaje sugestive. Programul va solicita numere până la apariţia unuia negativ, fapt care semnalează terminarea sa.