Proiect final\_2

## 1. Formulați o problemă din mediul real pentru care va temă sau algoritm. Impune restricții asupra variabilelor, tipurilor de date.

*Agricultorul Iurii are o livadă împărţită în n\*m sectoare* ***(2 < n< 100;2 < m < 100)****, dispuse pe n linii, numerotate de la 1 la n şi m coloane, numerotate de la 1 la m. În fiecare sector se află 3 copaci de nuci. După recoltarea nucilor el a comparat datele cu anii precedenţi, a observat că profitul, care îl aduce livada, an de an scade.*

*Pentru ameliorarea situaţiei el a hotărât să înlocuiască copacii, de pe sectoarele unde recolta e mai mică ca* ***valoarea medie pe livadă****, cu puieţi noi.*

*Pentru a rezolva problema el s-a adresat prietenului Ion, care este programator, pentru a-l ajuta la:*

* *calculul recoltei totale a livezii;*
* *calculul recoltei medii pe sector;*
* *calculul recoltei de pe sectoarele cu o recoltă* ***mai mică*** *ca recolta medie pe sector;*
* *calculul recoltei de pe sectoarele cu o recoltă* ***mai mare sau egală*** *cu recolta medie pe sector;*
* *numărarea şi identificarea sectoarelor cu o recoltă mai mică ca recolta medie pe sector.*

*Pentru a putea efectua calculul respectiv, Ion a solicitat următoarele date:*

* *dimensiunea livezii*
* *recolta de pe fiecare sector*

***NOTĂ****: Agricultorul estimează recolta în* ***q(chintale). 1q =100kg; 1kg=0,01q***

## 2. Explicaţi paşii de rezolvare a problemei.

* Calculăm recolta pe întreaga livadă;
* Calculăm recolta medie pe sector după formula: **;**
* Comparăm pe rând recolta de pe fiecare sector cu recolta medie,
  + selectăm din livadă sectoarele la care recolta este mai mică ca recolta medie pe sector;
  + numărăm acest sector;
  + calculăm suma recoltelor sectoarelor cu o recoltă mai mică ca recolta medie pe sector.

## 3. Descrieţi algoritmul prin pseudocod.

**START**

**INTRODU linii**

**INTRODU coloane**

**recolta=0**

**MERGEM DIN SECTOR ÎN SECTOR**

**SCRIE recolta = recolta + recolta\_pe\_sector[linie,coloana]**

**recolta\_media = recolta / (linii\*coloane)**

**MERGEM DIN SECTOR ÎN SECTOR**

**DACA sector[linie,coloana] < recolta\_media ATUNCI**

**SCRIE carnet(linie,coloana)**

**NUMARA sector\_rau**

**SUMA recolta\_sector\_rau**

**AFISEAZA recolta**

**AFISEAZA recolta\_medie**

**AFISEAZA recolta\_sector\_rau**

**AFISEAZA numarul\_sector\_rau**

**AFISEAZA coordonate\_sectoare\_rele**

**SFIRSIT**

## 4. Realizați schema bloc.

## 

## 5. Testarea algoritmul pentru valori de test.

1. linii - 3, coloane - 4

2 4 6 8

3 5 7 7

2 4 6 5

* *recoltei totale a livezii* ***59q;***
* *recoltei medii pe sector* ***4.916667q;***
* *recoltei de pe sectoarele cu o recoltă* ***mai mică*** *ca recolta medie pe sector* ***15q****;*
* *recoltei de pe sectoarele cu o recoltă* ***mai mare sau egală*** *cu recolta medie pe sector* ***44q****;*
* *numărul de sectoarelor cu o recoltă mai mică ca recolta medie pe sector* ***5 sectoare;***
* *sectoarelor cu o recoltă mai mică ca recolta medie pe sector:*

*sector****[1 , 1]****, recolta -* ***2q****;*

*sector****[1 , 2]****, recolta -* ***4q****;*

*sector****[2 , 1]****, recolta -* ***3q****;*

*sector****[3 , 1]****, recolta -* ***3q****;*

*sector****[3 , 2]****, recolta -* ***4q****;*

1. linii - 3, coloane - 3

1 2 3

4 5 6

7 8 9

* *recoltei totale a livezii* ***45q;***
* *recoltei medii pe sector* ***5q;***
* *recoltei de pe sectoarele cu o recoltă* ***mai mică*** *ca recolta medie pe sector* ***10q****;*
* *recoltei de pe sectoarele cu o recoltă* ***mai mare sau egală*** *cu recolta medie pe sector* ***35q****;*
* *numărul de sectoarelor cu o recoltă mai mică ca recolta medie pe sector* ***4 sectoare;***
* *sectoarelor cu o recoltă mai mică ca recolta medie pe sector:*

*sector****[1 , 1]****, recolta -* ***1q****;*

*sector****[1 , 2]****, recolta -* ***2q****;*

*sector****[1 , 3]****, recolta -* ***3q****;*

*sector****[1 , 2]****, recolta -* ***4q****;*

1. linii - 5, coloane - 5

2,5 3 5.1 6 9

5 8 7.5 3.2 8

4.32 5 6.25 3.1 4.8

5.25 7.5 5 2.2 1.9

8 5 6 7 3

* *recoltei totale a livezii* ***131.619995q;***
* *recoltei medii pe sector* ***5.26q;***
* *recoltei de pe sectoarele cu o recoltă* ***mai mică*** *ca recolta medie pe sector* ***58.37q****;*
* *recoltei de pe sectoarele cu o recoltă* ***mai mare sau egală*** *cu recolta medie pe sector* ***73.25q****;*
* *numărul de sectoarelor cu o recoltă mai mică ca recolta medie pe sector* ***15 sectoare;***
* *sectoarelor cu o recoltă mai mică ca recolta medie pe sector:*

*Sector****[1 , 1],*** *recolta* ***2.50q***

*Sector****[1 , 2],*** *recolta* ***3.00q***

*Sector****[1 , 3],*** *recolta* ***5.10q***

*Sector****[2 , 1],*** *recolta* ***5.00q***

*Sector****[2 , 4],*** *recolta* ***3.20q***

*Sector****[3 , 1],*** *recolta* ***4.32q***

*Sector****[3 , 2],*** *recolta* ***5.00q***

*Sector****[3 , 4],*** *recolta* ***3.10q***

*Sector****[3 , 5],*** *recolta* ***4.80q***

*Sector****[4 , 1],*** *recolta* ***5.25q***

*Sector****[4 , 3],*** *recolta* ***5.00q***

*Sector****[4 , 4],*** *recolta* ***2.20q***

*Sector****[4 , 5],*** *recolta* ***1.90q***

*Sector****[5 , 2],*** *recolta* ***5.00q***

*Sector****[5 , 5],*** *recolta* ***3.00q***

## 6. Realizați programul în limbajul C sau C++.

#include <stdio.h>

#include <iostream>

using namespace std;

#include <math.h>

#include <conio.h>

#define n\_jos 2

int linii, coloane, sector\_rau=0;

float recolta=0, recolta\_medie=0, recolta\_sector\_rau=0;

float sector[50][50];

struct blocnot{

int rind, coloana;

float recolta;

} carnet[50];

//--------------------!!! INTRODUCEREA !!!--------------------

void introdu\_linii()

{

do

{

printf(" Introdu numărul de rânduri a livezii ==> ");scanf("%d",&linii);

if (linii<=n\_jos)

{

system("clear");

printf(" Livada are mai mult de 2 râduri !!!");printf("\n");

printf(" Mai înceacă o dată !!!");printf("\n");

}

}

while (linii<=n\_jos);

}

//++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

void introdu\_coloane()

{

do

{

printf(" Introdu numărul de coloane a livezii ==> ");scanf("%d", &coloane);

if (coloane<=n\_jos)

{

system("clear");

printf(" Livada are mai mult de 2 coloane !!!");printf("\n");

printf(" Mai înceacă o dată !!!");printf("\n");

}

}

while (coloane<=n\_jos);

}

//++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

void introdu\_recolta()

{

for (int i = 0; i < linii; i++)

for (int j = 0; j < coloane; j++)

{

printf("sector[%d,%d] = ",i+1,j+1);

scanf("%f", &sector[i][j]);

}

return;

}

//--------------------!!! AFISAREA !!!--------------------

void afisare\_carnet()

{

for (int i = 0; i < sector\_rau; i++)

{

printf(" Sector [%d %d] ", carnet[i].rind+1, carnet[i].coloana+1);

printf(" - recolta %f q/sector",carnet[i].recolta);printf("\n");

}

return;

}

//--------------------!!! PRELUCRAREA datelor !!!--------------------

void calcul\_recolta()

{

for (int i = 0; i < linii; i++)

{

for (int j = 0; j < coloane; j++)

recolta+=sector[i][j];

printf("\n");

}

return;

}

//++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

void scrie\_carnet(int r, int c, float rr)

{

carnet[sector\_rau].rind=r;

carnet[sector\_rau].coloana=c;

carnet[sector\_rau].recolta=rr;

return;

}

//++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

void NUMARA\_sector\_rau(){ sector\_rau++;}

//++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

// PROGRAMUL PRINCIPAL

//++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

int main()

{

introdu\_linii();

system("clear");

introdu\_coloane();

introdu\_recolta();

system("clear");

calcul\_recolta();

recolta\_medie=recolta/(linii\*coloane);

for(int i=0;i< linii; i++)

{

for(int j=0; j<coloane;j++)

{

if(sector[i][j]<recolta\_medie)

{

scrie\_carnet(i,j,sector[i][j]);

NUMARA\_sector\_rau();

recolta\_sector\_rau+=sector[i][j];

}

}

}

system("clear");

printf("-------------------------------"); printf("\n");

printf(" Structura livezii:"); printf("\n");

printf("-------------------------------"); printf("\n");

printf(" - Rinduri = %d",linii); printf("\n");

printf(" - Coloane = %d",coloane); printf("\n");

printf(" - Sectoare = %d",linii\*coloane); printf("\n");

printf(" - Copaci = %d",3\*linii\*coloane); printf("\n");

printf("-------------------------------"); printf("\n");

printf(" Recolta totala = %f q",recolta); printf("\n");

printf("-------------------------------"); printf("\n");

printf(" - Recolta medie = %f q/sector",recolta\_medie); printf("\n");

printf(" - Recolta sector\_rau = %f q/sector",recolta\_sector\_rau); printf("\n");

printf(" - Recolta sector\_bun = %f q/sector",recolta-recolta\_sector\_rau); printf("\n");

printf("-------------------------------"); printf("\n");

printf(" sectoar\_rau = %d sectoare",sector\_rau); printf("\n");

printf(" Sectoare cu o recoltă mai mică ca recolta medie pe sector : "); printf("\n");

printf("-------------------------------"); printf("\n");

afisare\_carnet();

return 1;

}

## 7. Prezentaţi variante de valori iniţiale (baterii de teste - trei teste).

1. linii - 3, coloane - 4

2 4 6 8

3 5 7 7

2 4 6 5

1. linii - 3, coloane - 3

1 2 3

4 5 6

7 8 9

1. linii - 5, coloane - 5

2,5 3 5.1 6 9

5 8 7.5 3.2 8

4.32 5 6.25 3.1 4.8

5.25 7.5 5 2.2 1.9

8 5 6 7 3

## 8. Realizaţi un barem de evaluare pentru cazul când programul elaborat nu lucrează pentru bateriile de teste propuse.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Motivatrea punctelor** | **Puncte** | **Total** |
| **1** | **Structură program** |  |  |
| funcţia main | 1P |
|  |  |
| **2** | **Declaraţii** |  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **3** | **Introducerea datelor** |  |  |
|  |  |
|  |  |
|  |  |
| **4** | **Prelucrarea datelor** |  |  |
|  |  |
|  |  |
| **5** | **Afişarea rezultatelor** |  |  |
|  |  |
|  |  |