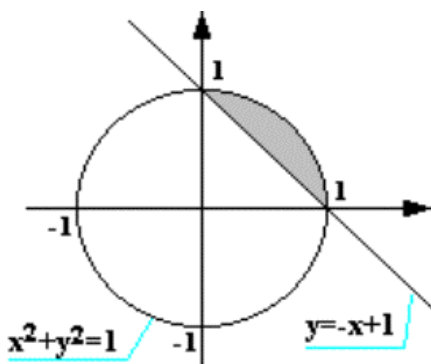


**Tema: Aplicarea instrucțiunilor decizionale (if)****Obiective:**

- să definească noțiunea de instrucțiune decizională;
- să recunoască contextul utilizării instrucțiunilor decizionale;
- să descrie și aplice diferite metode de organizare a ramificărilor în programe;
- să propună algoritmi simpli de soluționare a problemelor.

**5.1. Sarcină pentru realizare**

De elaborat un program care citește coordonatele punctului  $(x, y)$  și determină, dacă punctul este amplasat în domeniul hașurat din figura 5.1. Amplasarea pe hotarul domeniului să fie considerată nimerirea în domeniu.



**Figura 5.1.** Ecuațiile ce descriu domeniul hașurat

**5.2. Exemplu de soluționare a sarcinii****Elaborarea algoritmului de calcul**

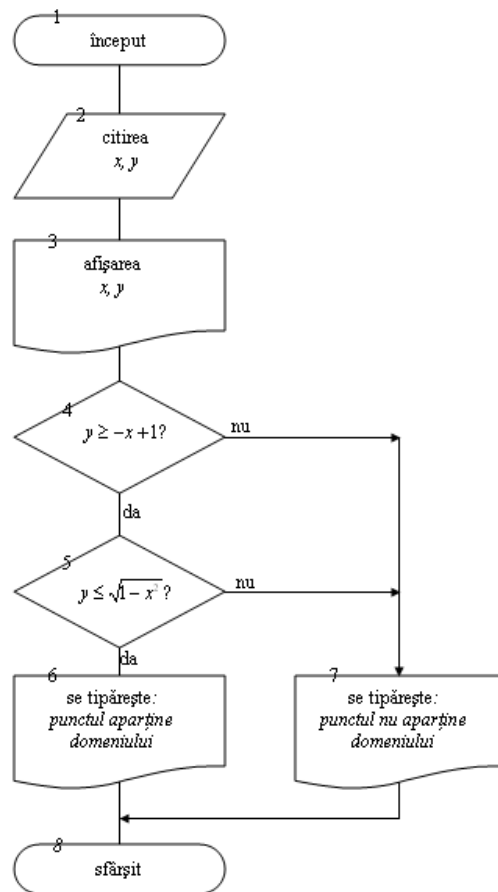
Primul pas al algoritmului îl constituie introducerea coordonatelor punctului:  $x$  și  $y$ . Pentru o comoditate mai mare la analiza rezultatelor, mărimile introduse pot fi afișate la ecran.

Conform imaginii din Figura 5.1, se poate de menționat că domeniul se formează la intersecția drepte descrise de ecuația  $y = -x + 1$  și circumferința  $x^2 + y^2 = 1$ .

Pentru a stabili dacă punctul aparține domeniului, este necesar de verificat următoarele condiții:

- 1) punctul este amplasat mai sus sau pe dreaptă:  $y \geq -x + 1$  ;
- 2) punctul este amplasat în circumferință sau pe marginea ei:  $x^2 + y^2 \leq 1$ .

Punctul aparține domeniului hașurat, dacă se îndeplinesc ambele condiții. În caz că nu se satisface măcar una dintre aceste condiții (a se vedea Figura 5.2, blocul 4,5,7), punctul nu aparține domeniului. Prin urmare, aceste condiții urmează a fi unite cu ajutorul operatorului logic  $\&\&$ .



**Figura 5.2.** Schema bloc

### Determinarea variabilelor programului

Pentru realizarea algoritmului sunt necesare doar variabile pentru păstrarea coordonatelor  $x$  și  $y$ . În condițiile sarcinii nu sunt aduse cerințe pentru precizia calculelor, desenul din Figura 5.1 este prezentat cu o precizie suficient de mică, pentru variabile tipul *float* va fi de ajuns. Dar în corespundere cu stilul comun de programare în C se alege pentru ele tipul *double*.

### Elaborarea textului programului

Textul programului începe cu includerea fișierului antet:

```
#include <stdio.h>
```

În continuare, urmează antetul și deschiderea funcției principale:

```
int main(void) {
```

declararea variabilelor:

```
double x, y;
```

Pentru fiecare coordonată, se afișează invitația pentru introducerea ei și citirea valorii:

```
printf("Introduceți coordonata x >");
```

```
scanf("%lf",&x);
```

```
printf("Introduceți coordonata y >");
```

```
scanf("%lf",&y);
```

Valorile coordonatelor se extrag la ecran:

```
printf("x=%6.3lf; y=%6.3lf\n",x,y);
```

În continuare, se verifică dacă punctul aparține domeniului hașurat. Ambele condiții se verifică într-o singură expresie. Pentru ca ambele condiții să se satisfacă, în expresie sunt unite cu operatorul logic &&:

```
if ( (y>=1-x)&& (x*x+y*y<=1) )
```

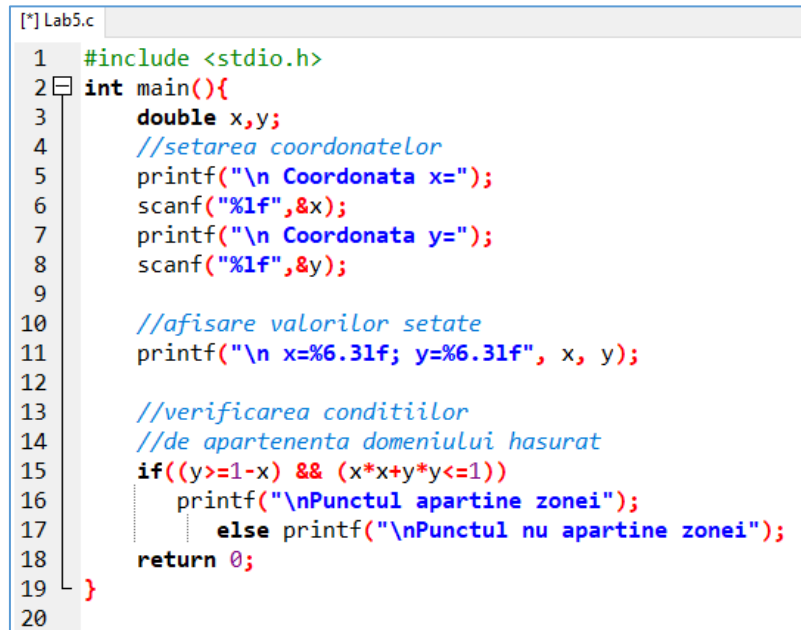
Dacă valoarea expresiei logice în instrucțiunea condițională este TRUE, atunci se va afișa mesajul de apartenență:

```
printf("Punctul aparține zonei\n");
```

În caz contrar, se afișează mesajul corespunzător:

```
else printf("Punctul nu aparține zonei\n");
```

Textul integral al programului este prezentat în Figura 5.3.



```
[*] Lab5.c
1  #include <stdio.h>
2  int main(){
3      double x,y;
4      //setarea coordonatelor
5      printf("\n Coordonata x=");
6      scanf("%lf",&x);
7      printf("\n Coordonata y=");
8      scanf("%lf",&y);
9
10     //afisare valorilor setate
11     printf("\n x=%6.3lf; y=%6.3lf", x, y);
12
13     //verificarea conditiilor
14     //de apartenenta domeniului hasurat
15     if((y>=1-x) && (x*x+y*y<=1))
16         printf("\nPunctul apartine zonei");
17     else printf("\nPunctul nu apartine zonei");
18     return 0;
19 }
20
```

**Figura 5.3.** Textul programului

### Ajustarea programului

Cel mai important moment, în ajustarea programului, constă în a vă convinge că programul emite rezultate corecte pentru puncte cu diferite coordonate. Prin urmare, este necesar de selectat astfel de combinații, care vor fi demonstrative pentru diferite cazuri de amplasare a punctului. Noi propunem astfel de combinații:

- 1) punctul este amplasat mai jos de zonă – (0.4, 0.4);
- 2) punctul este amplasat pe hotarul de jos a zonei – (0.5, 0.5);
- 3) punctul este amplasat în zonă – (0.7, 0.6);
- 4) punctul este amplasat pe hotarul de sus – (0.707, 0.707);
- 5) punctul este amplasat mai sus de zonă – (0.8, 0.8);
- 6) punctul este amplasat pe marginea din dreapta zonei – (1.0, 0.0);
- 7) punctul este amplasat pe marginea din stânga zonei – (0.0, 1.0);

### Rezultatele derulării programului

Mai jos sunt aduse rezultatele lucrului programului:

```

Introduceti coordonata x >0.7
Introduceti coordonata y >0.6
x= 0.700; y= 0.600
Punctul apartine zonei
Introduceti coordonata x >0.707
Introduceti coordonata y >0.707
x= 0.707; y= 0.707
Punctul apartine zonei
Introduceti coordonata x >0.8
Introduceti coordonata y >0.8
x= 0.800; y= 0.800
Punctul nu apartine zonei
Introduceti coordonata x >0.1
Introduceti coordonata y >0.1
x= 0.100; y= 0.100
Punctul nu apartine zonei
Introduceti coordonata x >0
Introduceti coordonata y >1
x= 0.000; y= 1.000
Punctul apartine zonei

```

**Figura 5.4.** Rezultatele furnizate de program

### 5.3. Subiecte de evaluare

#### Exerciții:

1. Să se implementeze programul din exemplul prezentat.
2. Identificați valorile returnate de următoarele secvențe de program:

```

main() {
    int x, y=1, z;
    if ( y!=0 ) x=5;
    printf("\nx=%d",x);          //1

    if (y==0) x=3;
    else x=5;
    printf("\nx=%d",x);          //2

    x=1;
    if (y<0) if (y>0) x=3;
    else x=5;
    printf("\nx=%d",x);          //3

    if ( z=y<0 ) x=3;
    else if (y==0) x=5;
    else x=7;
    printf("\nx=%d z=%d",x,d);    //4

    if (z=(y==0)) x=5; x=3;
    printf("\nx=%d z=%d",x,d)     //5

    if (x=z=y); x=3;
    printf("\nx=%d z=%d",x,d); }  //6

```

**Verificarea cunoștințelor:**

1. Definiți noțiunea de instrucțiune.
2. Identificați necesitatea utilizării instrucțiunii *if*.
3. Descrieți formele de înscriere a instrucțiunii *if*.
1. Prezentați exemple de aplicare a instrucțiunii *if* plasate.
4. Definiți noțiunea de instrucțiune compusă.
5. Identificați cazurile recomandate aplicării operatorilor logice ȘI (&&) , SAU (||) în instrucțiunea *if*.
6. Enumerați deosebirile de bază dintre instrucțiunea condițională și operatorul condițional.

**5.4. Bibliografie**

1. Kris Jamsa & Lars Klander, *Totul despre C și C++*, Editura Teora, București, 2006.
2. Herbert Schildt, *C++ manual complet*, Editura Teora, București, 1999. Disponibil: <https://drive.google.com/file/d/1BrQtITgykcWk03xxtl3Q0-nvcRRFAsy7/view?usp=sharing>
3. Tutorials C++ Language. Operators. Disponibil: <https://cplusplus.com/doc/tutorial-ro/control/>