МІНІСТЕРСТВО ОСВІТИ І НАУКИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра систем штучного інтелекту



Лабораторна робота №3

З курсу “Дискретна математика ”

Виконав:  
ст.гр. КН-110

Пиріг Богдан

Викладач:

Мельникова Н.І.

Львів – 2018

**Тема: Побудова матриці бінарного відношення**

**Мета роботи: набуття практичних вмінь та навичок при побудові матриць бінарних відношень та визначені їх типів**

**Варіант №6  
  
Завдання №1**

****

Нехай (x, y) ∈ (A x B) ∩(C x D) ⬄

(x, y) ∈ (A x B) & (x, y) ∈ (C x D)⬄

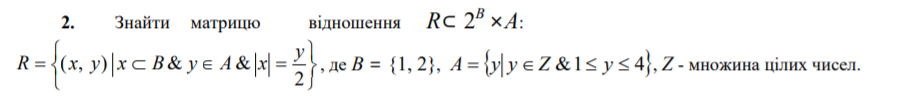
(x ∈ A & y ∈ B) & (x ∈ C & y ∈ D) ⬄

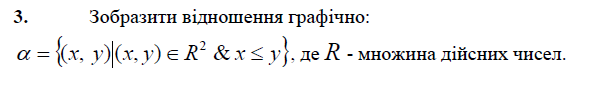
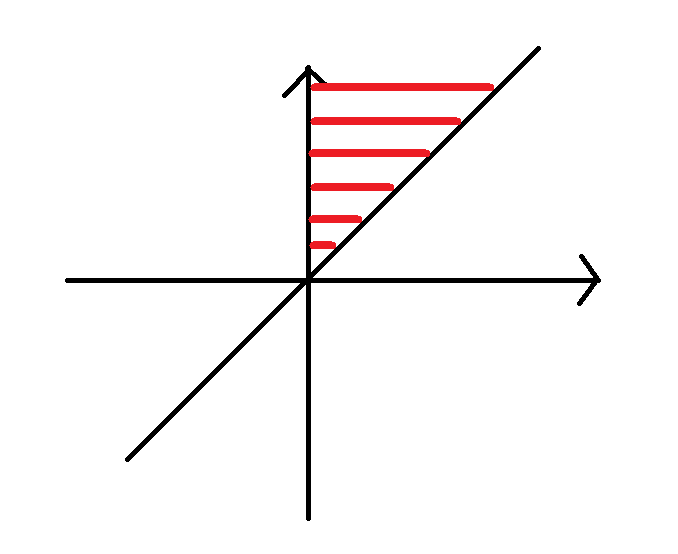
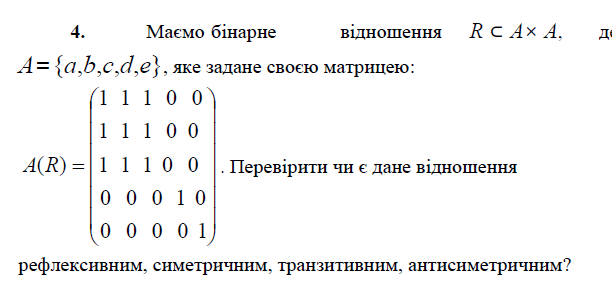
(x ∈ A & x ∈ C) & (y ∈ B & y ∈ D) ⬄

(x ∈ A x C) & (y ∈ B x D) ⬄ (x, y) ∈ (A x C)∩(B x D)

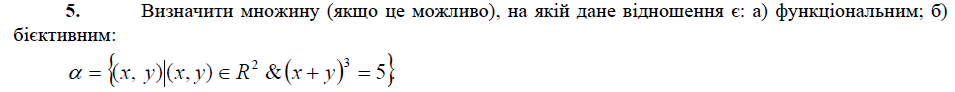
Отже, ця рівність є вірною.

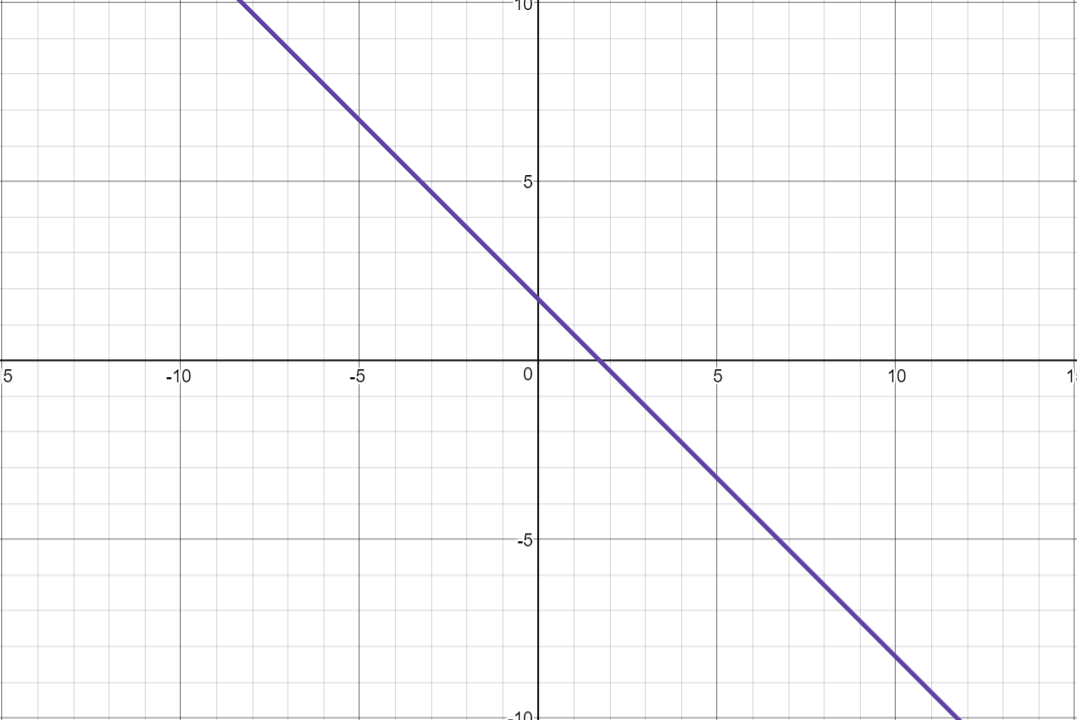
**Завдання №2**

****

**Завдання №3  
  
Завдання №4**

Дане відношення є рефлексивним, адже головна діагональ матриці складається з одиниць, симетричним, тому що матриця симетрична відносно головної діагоналі, і не є транзитивним.

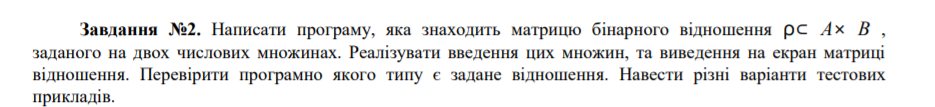
**Завдання №5  
  
**

  
Кожному елементу x ∈ A відповідає один і тільки один елемент y ∈ B і кожному елементу y ∈ B відповідає один і тільки один елемент x ∈ A. Ця функція є взаємооднозначною.

Функція, що визначає взаємнооднозначну відповідність називається бієктивною функцією.

Якщо функція бієктивна, то вона є функціональною.

**Завдання №6**

  
  
  
**Програма :**

1. **#include <stdio.h>**
2. **#include <stdlib.h>**
3. **#include <cs50.h>**
4. **int main ()**
5. **{**
6. **int n;**
7. **do**
8. **{**
9. **printf ("Size of massives 1 and 2 (only positive)- ");**
10. **n=GetInt();**
11. **if (n<=0) printf ("Try again\n");**
12. **}**
13. **while (n<=0);**
14. **int a[n];**
15. **printf ("Input your massive 1 (each element <10)\n");**
16. **for (int i=0; i<n; i++)**
17. **{**
18. **do**
19. **{**
20. **printf ("a[%d]=",i);**
21. **a[i]=GetInt();**
22. **if (a[i]>10 || a[i]<0) printf ("Element must be >0 and <10\n");**
23. **}**
24. **while (a[i]>10 || a[i]<0);**
25. **}**
26. **printf ("\nYour massive 1 : ");**
27. **for (int i=0; i<n; i++)**
28. **{**
29. **printf ("%d ", a[i]);**
30. **}**
31. **int b[n];**
32. **printf ("\nInput your massive 2 (each element <10)\n");**
33. **for (int i=0; i<n; i++)**
34. **{**
35. **do**
36. **{**
37. **printf ("b[%d]=",i);**
38. **b[i]=GetInt();**
39. **if (b[i]>10 || b[i]<0) printf ("Element must be >0 and <10\n");**
40. **}**
41. **while (b[i]>10 || b[i]<0);**
42. **}**
43. **printf ("\nYour massive 2 : ");**
44. **for (int i=0; i<n; i++)**
45. **{**
46. **printf ("%d ", b[i]);**
47. **}**
48. **printf ("\n");**
49. **int c[n][n];**
50. **for (int i=0; i<n; i++){**
51. **for (int j=0; j<n; j++)**
52. **{**
53. **if (2\*a[i]<b[j])**
54. **c[i][j]=1;**
55. **else if (2\*a[i]>b[j])**
56. **c[i][j]=0;**
57. **else if (2\*a[i]==b[j])**
58. **c[i][j]=0;**
59. **}**
60. **}**
61. **printf ("\nYour matrix of binary relation :\n");**
62. **for (int i=0; i<n; i++){**
63. **for (int j=0; j<n; j++)**
64. **{**
65. **printf ("%d ",c[j][i]);**
66. **}**
67. **printf ("\n");**
68. **}**
69. **int p=0;**
70. **int h=0;**
71. **for (int i=0; i<n; i++)**
72. **{**
73. **if (c[i][i]==1)**
74. **p++;**
75. **else if (c[i][i]==0)**
76. **h++;**
77. **}**
78. **if (p==n) printf ("Your matrix - reflective\n");**
79. **else if (p!=n && h!=n) printf ("Your matrix - not reflective\n");**
80. **else if (h==n) printf ("Your matrix anti-reflective\n");**
81. **int r=1;**
82. **int t=1;**
83. **for (int i=0; i<n; i++){**
84. **for (int j=0; j<n; j++){**
85. **for (int k=0; k<n; k++)**
86. **{**
87. **if (c[i][j] && c[j][k] && c[i][k] && (i!=k) && (i!=j))**
88. **r=0;**
89. **else if (c[i][j] && c[j][k] && !c[i][k] && (i!=k) && (i!=j))**
90. **t=0;**
91. **}**
92. **}**
93. **}**
94. **if (((r==1) && (t==1))||(r==0 && t==0)) printf ("Your matrix - not tranzitive\n");**
95. **else if (t==1) printf ("Your matrix - tranzitive\n");**
96. **else if (r==1) printf ("Your matrix - anti-tranzitive\n");**
97. **int l=1;**
98. **int k=0;**
99. **for (int i=0; i<n; i++){**
100. **for (int j=0; j<n; j++)**
101. **{**
102. **if (i!=j && c[i][j]!=c[j][i] && c[i][j]==1)**
103. **l=0;**
104. **else if (i!=j && c[i][j]==c[j][i] && c[i][j]==1)**
105. **k++;**
106. **}**
107. **}**
108. **if (l==1 && k>0) printf ("Your matrix - symmetric\n");**
109. **else if (l==0 && k==0) printf ("Your matrix - anti-symmetric\n");**
110. **else printf ("Your matrix - not symmetric\n");**
111. **return 0;**
112. **}**

**Результат :**

Size of massives 1 and 2 (only positive)- 2

Input your massive 1 (each element <10)

a[0]=1

a[1]=2

Your massive 1 : 1 2

Input your massive 2 (each element <10)

b[0]=3

b[1]=4

Your massive 2 : 3 4

Your matrix of binary relation :

1 0

1 0

Your matrix - not reflective

Your matrix - not tranzitive

Your matrix - anti-symmetric