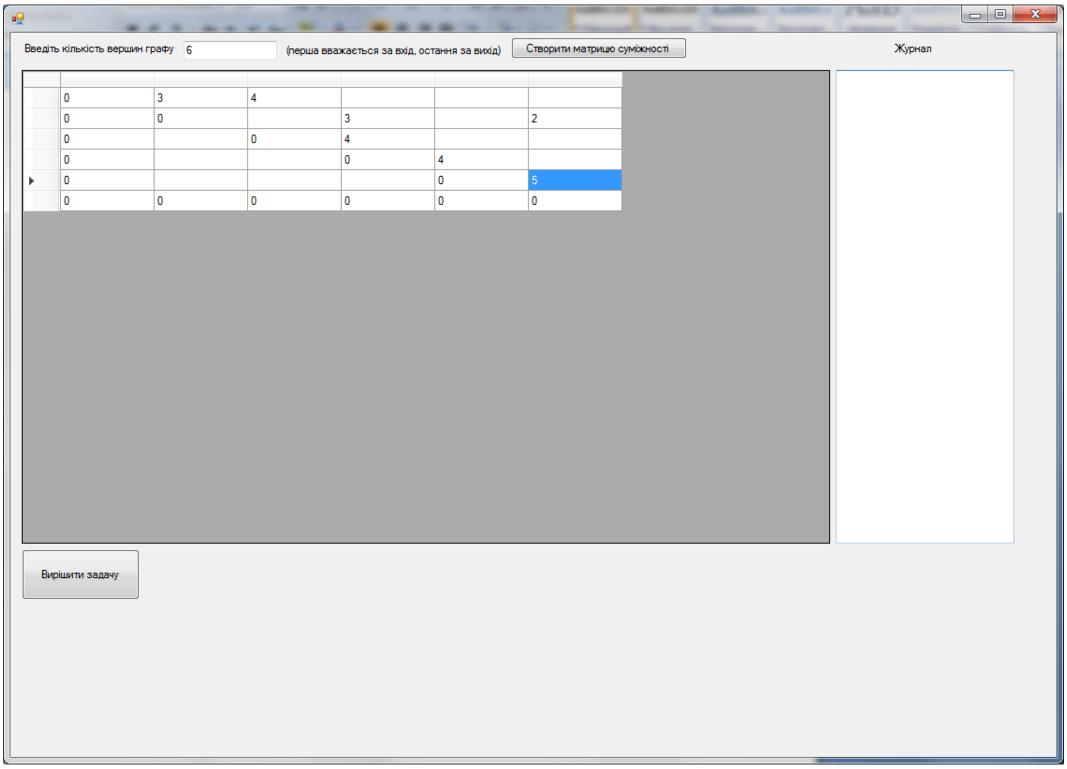
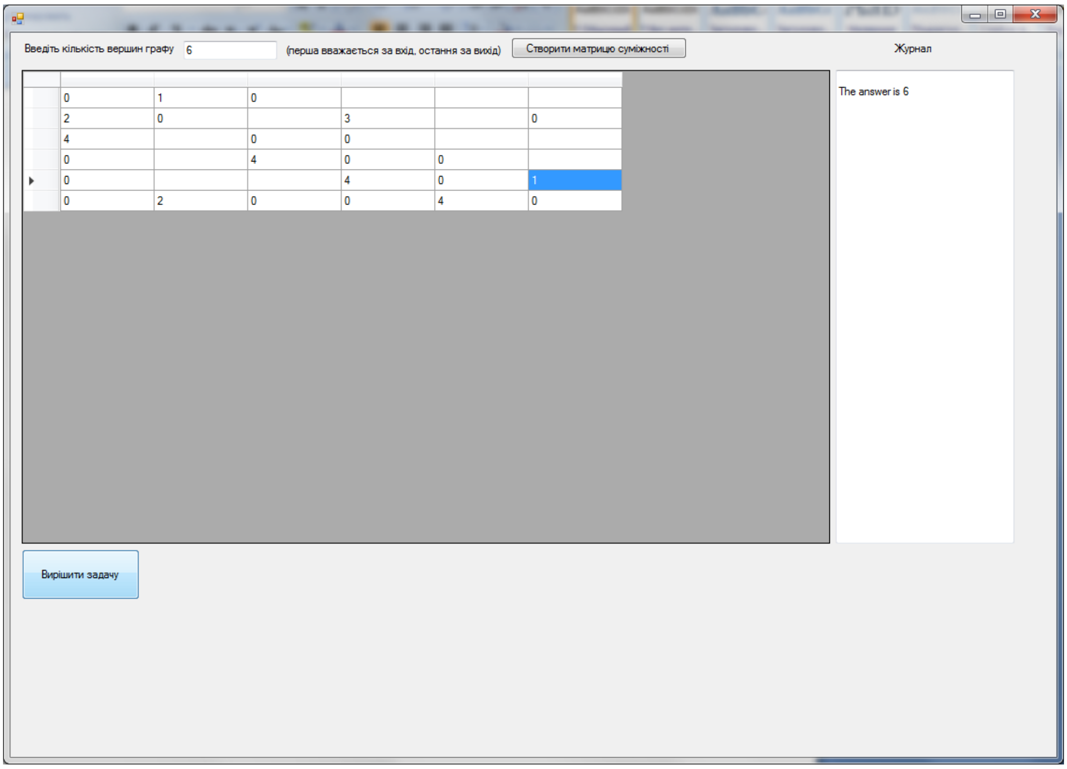


Введіть кількість вершин графу 6 (перша вважається за вхід, остання за вихід)

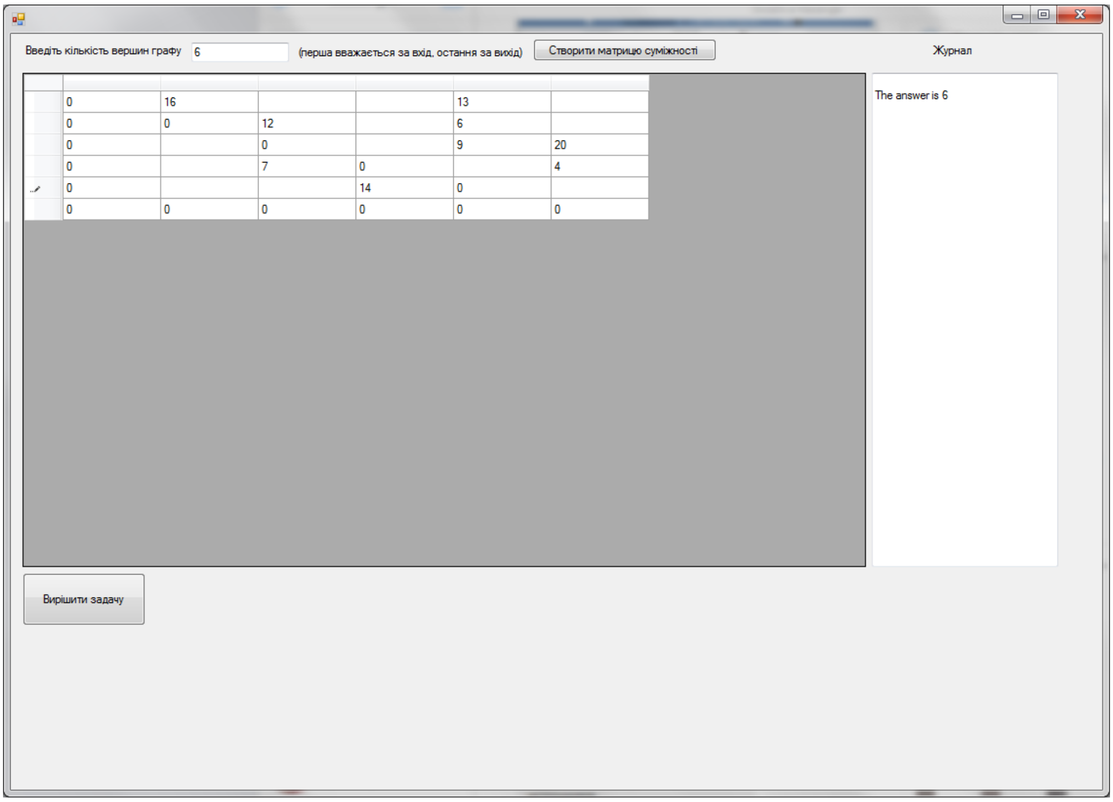
0	3	4			
0	0		3		2
0		0	4		
0			0	4	
0				0	5
0	0	0	0	0	0



Введіть кількість вершин графу 6 (перша вважається за вхід, остання за вихід)

0	1	0			
2	0		3		0
4		0	0		
0		4	0	0	
0			4	0	1
0	2	0	0	4	0

The answer is 6

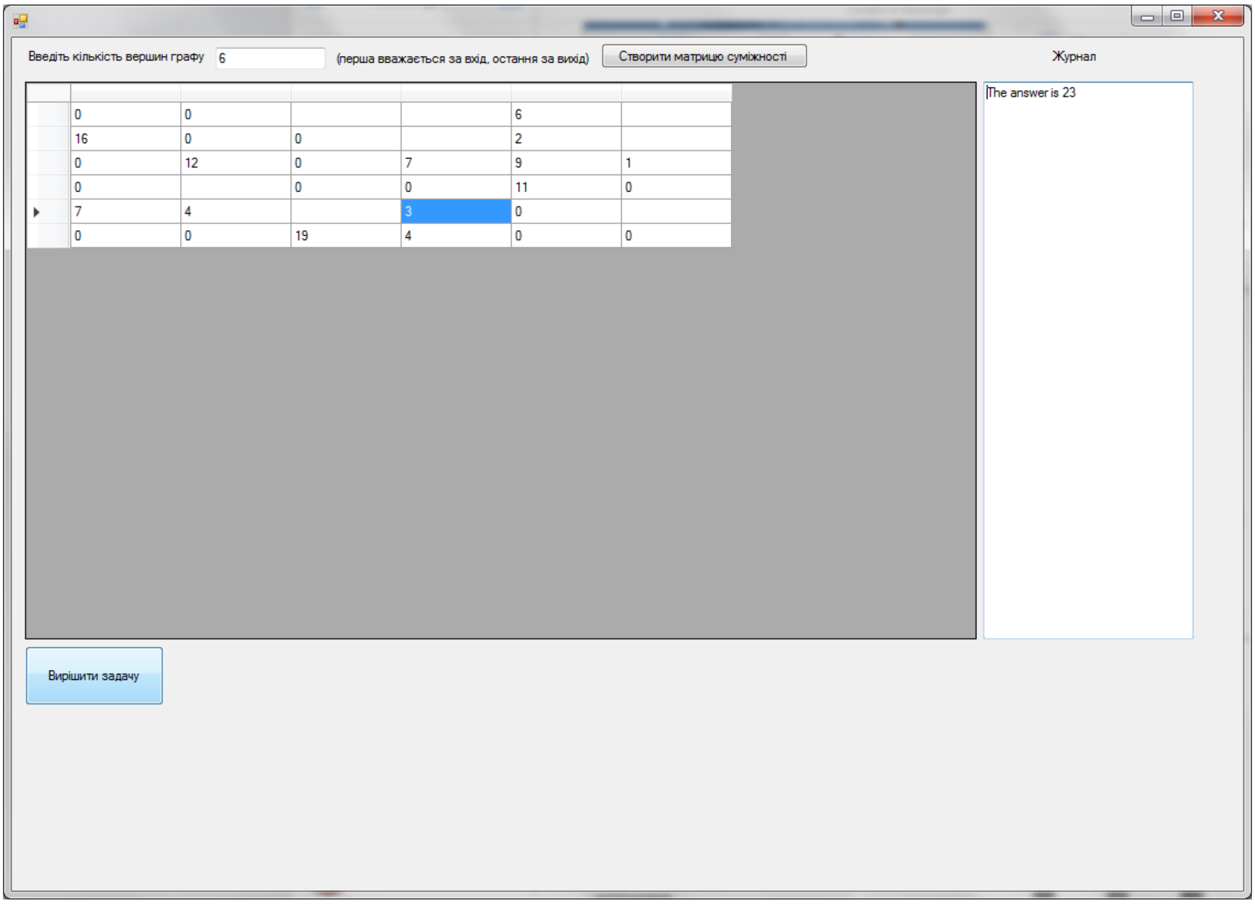


Введіть кількість вершин графу 6 (перша вважається за вхід, остання за вихід)

Журнал

0	16			13	
0	0	12		6	
0		0		9	20
0		7	0		4
0			14	0	
0	0	0	0	0	0

The answer is 6



Введіть кількість вершин графу 6 (перша вважається за вхід, остання за вихід)

Журнал

0	0			6	
16	0	0		2	
0	12	0	7	9	1
0		0	0	11	0
7	4		3	0	
0	0	19	4	0	0

The answer is 23

Код програми:

```
             if(dataGridView1.Columns.Count<=1)
            {
                textBox3.Text += "You shall create graph" + Environment.NewLine;
                return;
            }
            int i,j;
            int NumOfVec = dataGridView1.Columns.Count;
            int Check;
            try
            {
                for (i = 0; i < NumOfVec; i++)
                {
                    for (j = 0; j < NumOfVec; j++)
                    {
                        if(dataGridView1.Rows[i].Cells[j].ReadOnly==false)
                        {
                            Check = 
Convert.ToInt32(dataGridView1.Rows[i].Cells[j].Value);
                            if (Check < 0)
                                throw new Exception();
                        }
                    }
                }
            }
            catch
            {
                textBox3.Text += "Wrong input" + Environment.NewLine;
                return;
            }
            List<Way> CurWays;
            List<Way> LastWays;
            int AllStream = 0;
            bool StepMade;
            bool MoveMade = true;
            bool DeleteNeeded = false;
            int Temp;
            while(MoveMade)
            {
                StepMade = true;
                MoveMade = false;
                CurWays = new List<Way>();
                LastWays = new List<Way>();
                LastWays.Add(new Way(0));
                while(StepMade)
                {
                    CurWays = new List<Way>();
                    //CurWays = LastWays;
                    StepMade = false;
                    for (i = 0; i < LastWays.Count; i++)
                    {
                        //DeleteNeeded = true;
                        if(LastWays[i].CurWay[LastWays[i].CurWay.Count-
1]==(dataGridView1.Rows.Count-1))
                        {
                            CurWays.Add(LastWays[i]);
                            //StepMade = false;
                            continue;
                           // DeleteNeeded = false;
                        }
                        for (j = 0; j < dataGridView1.ColumnCount; j++)
                        {
                            if 
(Convert.ToInt32(dataGridView1.Rows[LastWays[i].CurWay[LastWays[i].CurWay.Count - 
1]].Cells[j].Value) != 0)
```

```
                            {
                                CurWays.Add(new Way(LastWays[i]));
                                CurWays[CurWays.Count -
1].AddPoint(j,Convert.ToInt32(dataGridView1.Rows[LastWays[i].CurWay[LastWays[i].CurWay.Co
unt - 1]].Cells[j].Value));
                                DeleteNeeded = false;
                                StepMade = true;
                                if (CurWays[CurWays.Count - 1].CheckCycle())
                                {
                                    CurWays.RemoveAt(CurWays.Count - 1);
                                }
                            }
                        }
                        /*if (DeleteNeeded)
                        {
                            LastWays.RemoveAt(i);
                            i--;
                        }*/
                    }
                    LastWays = CurWays;
                }
                LastWays.Sort();
                if (LastWays.Count!=0 && (LastWays[LastWays.Count - 1].MinVal != 0  ||
LastWays[LastWays.Count - 1].MinVal != int.MaxValue ))
                {
                    AllStream+=LastWays[LastWays.Count - 1].MinVal;
                    for (i = 1; i < LastWays[LastWays.Count - 1].CurWay.Count;i++ )
                    {
                        dataGridView1.Rows[LastWays[LastWays.Count - 1].CurWay[i -
1]].Cells[LastWays[LastWays.Count - 1].CurWay[i]].Value =
Convert.ToInt32(dataGridView1.Rows[LastWays[LastWays.Count - 1].CurWay[i -
1]].Cells[LastWays[LastWays.Count - 1].CurWay[i]].Value) - LastWays[LastWays.Count -
1].MinVal;
                        dataGridView1.Rows[LastWays[LastWays.Count -
1].CurWay[i]].Cells[LastWays[LastWays.Count - 1].CurWay[i - 1]].Value =
Convert.ToInt32(dataGridView1.Rows[LastWays[LastWays.Count -
1].CurWay[i]].Cells[LastWays[LastWays.Count - 1].CurWay[i-1]].Value) +
LastWays[LastWays.Count - 1].MinVal;
                    }
                    MoveMade = true;
                }
            }
            textBox3.Text += "The answer is " + AllStream + Environment.NewLine;
```