**Problem---1**

int b,c,d;

string numbersLine = Console.ReadLine();

string[] numbers = numbersLine.Split(new char[] { ' ' });

Console.WriteLine(numbers[0]);

Console.WriteLine(numbers[1]);

b = Convert.ToInt32(numbers[0]);

c = Convert.ToInt32(numbers[1]);

if (1<=c & c <= 30)

{

d = (b << c) | (b >> (32 - c));

Console.WriteLine(Convert.ToString(b, toBase: 2));

Console.WriteLine(Convert.ToString(d, toBase: 2));

int e= Convert.ToInt32(d);

Console.WriteLine(e);

else {

Console.WriteLine("Error");

}

**2. Problem---2**

char [] a = new char[100];

char[] d = new char[100];

int[] b = new int[100];

int[] c = new int[100];

Console.WriteLine("Enter string digits up to 100 split by space");

int i,sum=0;

string array1 = Console.ReadLine();

string[] list1 = array1.Split(" ");

int largesizeaerray;

Console.WriteLine("Enter string digits up to 100 split by space");

string array2 = Console.ReadLine();

string[] list2 = array2.Split(" ");

Console.WriteLine("First Line Integer");

for ( i = 0; i < list1.Length; i++) {

a[i] = (Convert.ToChar(list1[i]));

b[i] = (int)(Convert.ToChar(list1[i]));

Console.WriteLine(b[i]);

}

Console.WriteLine("Second Line Integer");

for (i = 0; i < list2.Length; i++)

{

d[i] = (Convert.ToChar(list2[i]));

c[i] = (int)(Convert.ToChar(list2[i]));

Console.WriteLine(c[i]);

}

if (list1.Length >= list2.Length)

{

largesizeaerray = list1.Length;

}

else {

largesizeaerray = list2.Length;

}

string t = String.Format("--SUM----");

Console.WriteLine(t);

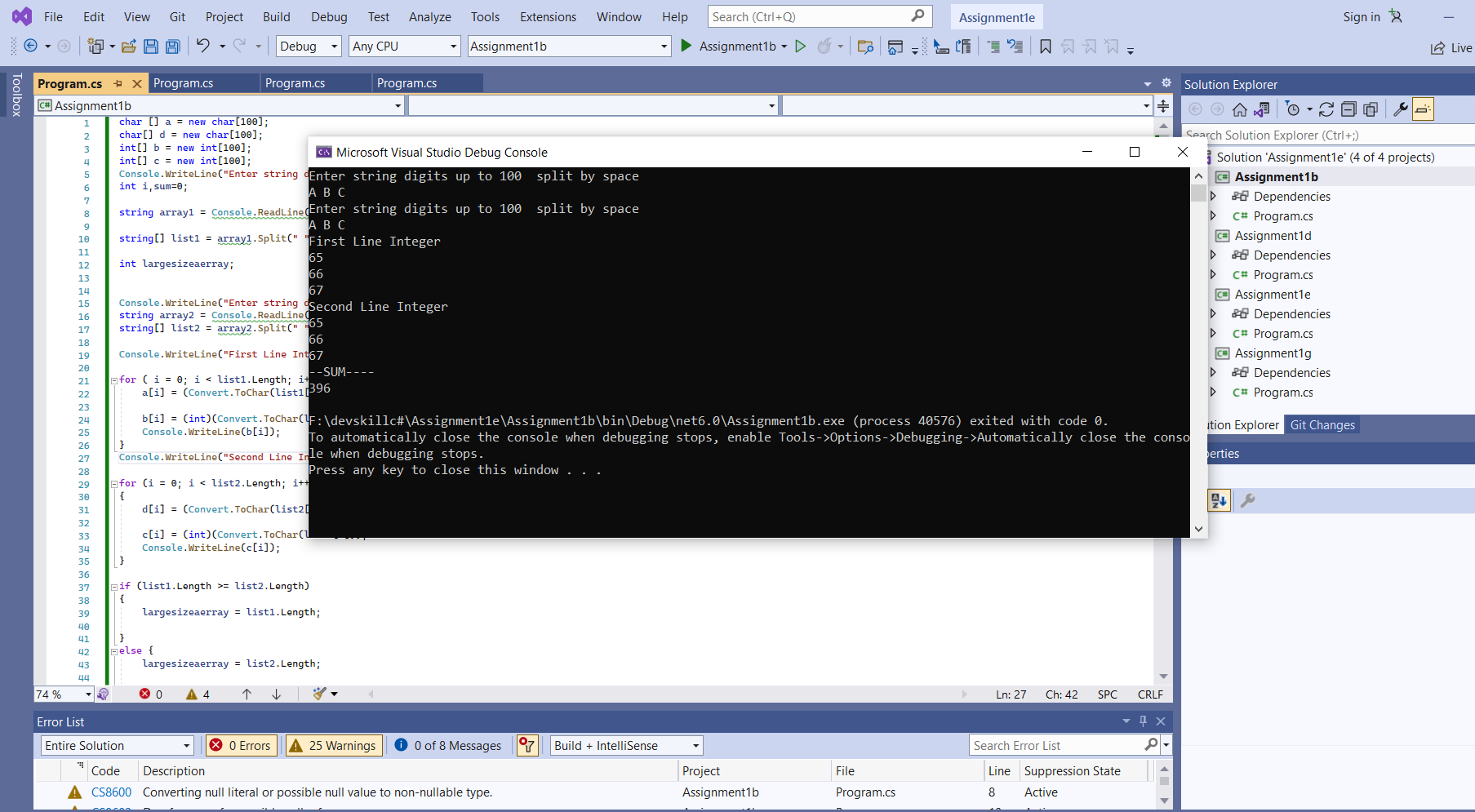
for (i = 0; i < largesizeaerray; i++)

{

sum = sum + b[i] + c[i];

}

Console.WriteLine(sum);



**Problem-3**

int[][,] jaggedArray4 = new int[][,]

{

new int[,] { {1,3}, {5,7} },

new int[,] { {0,2}, {4,6}, {8,10} },

new int[,] { {11,22}, {99,88}, {0,9} }

};

int[] c = new int[30];

int d;

for (int i = 0; i < 3; i++) {

for (int j = 0; j <3 ; j++)

{

for (int k = 0; k <2; k++)

{

if (i == 0 && j == 2)

{

}

else {

d = jaggedArray4[i][j, k];

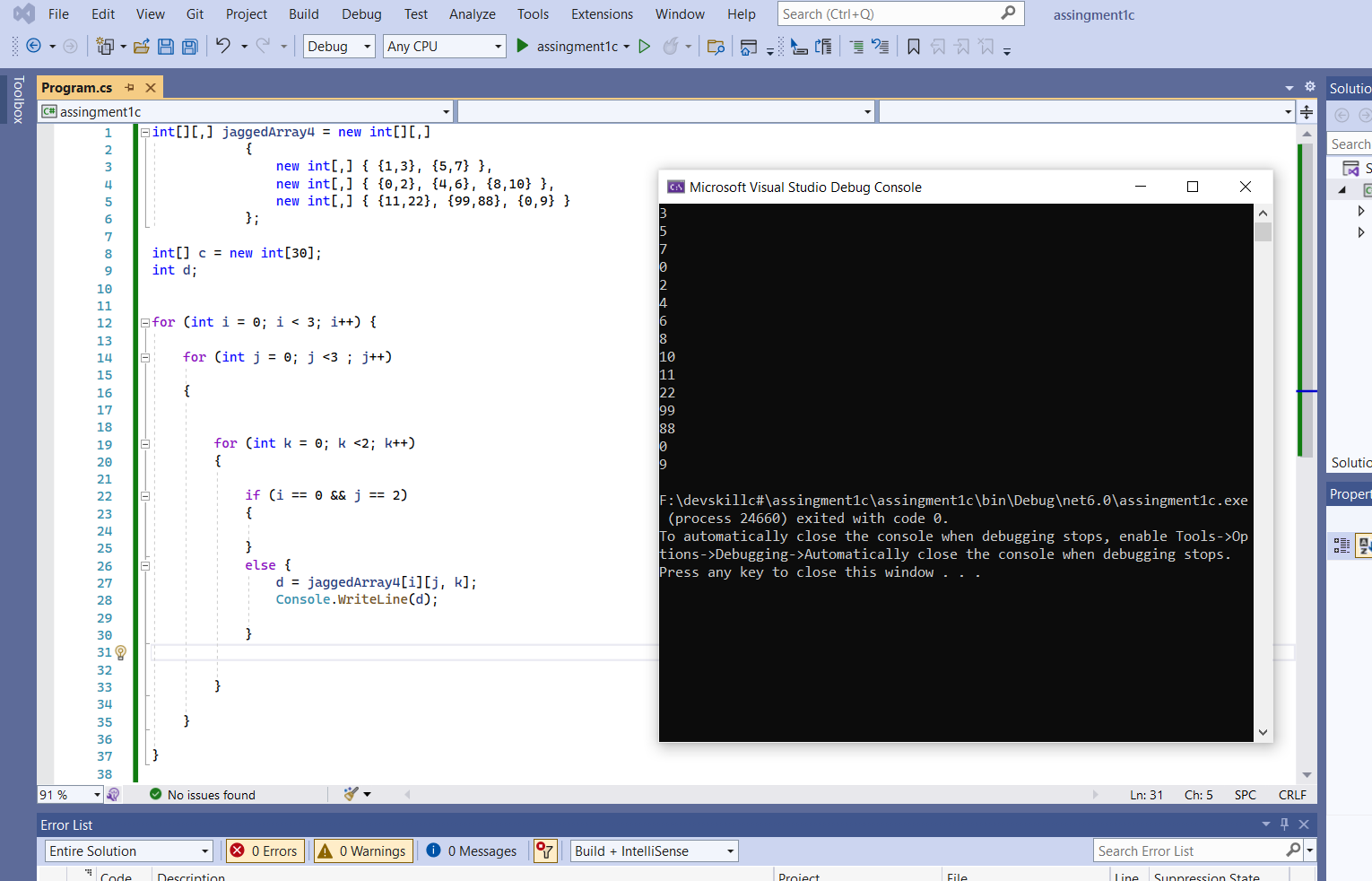
Console.WriteLine(d);

}

}

}

}



**Problem-4**

int[] a = new int[10];

int[] b = new int[10];

int[] c = new int[10];

int[] d = new int[10];

int[] cdsize = new int[2];

int[] e = new int[2];

int[] revc = new int[10];

int[] revd = new int[10];

int clen, g, h;

int dlen;

Console.WriteLine("Enter digits up to 9 with split by space");

string arrayTemp1 = Convert.ToString(Console.ReadLine());

string[] temp1 = arrayTemp1.Split(' ');

for (int i = 0; i < temp1.Length; i++)

{

a[i] = Convert.ToInt32(temp1[i]);

}

string t = String.Format("a. Reverse the order of the number =");

Console.WriteLine(t);

for (int j = (temp1.Length) - 1, k = 1; j >= 0; j--, k++)

{

b[k] = a[j];

Console.WriteLine(b[k]);

}

int n = 1, p = 1;

for (int m = 1; m <= temp1.Length; m++)

{

if (m % 2 == 0)

{

c[n] = b[m];

n++;

}

else

{

d[p] = b[m];

p++;

}

}

Console.WriteLine("c-Even array-");

for (int m = 1; m < c.Length; m++)

{

if (c[m] == 0)

{

cdsize[0] = m - 1;

break;

}

Console.WriteLine(c[m]);

}

Console.Write("c-Even size-");

Console.WriteLine(cdsize[0]);

Console.WriteLine("D-odd array-");

for (int m = 1; m < c.Length; m++)

{

if (d[m] == 0)

{

cdsize[1] = m - 1;

break;

}

Console.WriteLine(d[m]);

}

Console.Write("D-Odd size-");

Console.WriteLine(cdsize[1]);

for (int m = 1; m < c.Length; m++)

{

if (c[m] == 0)

{

cdsize[1] = m - 1;

break;

}

}

Console.WriteLine("----------c--------------");

for (int j = cdsize[0], k = 1; j > 0; j--, k++)

{

revc[k] = c[j];

Console.Write(revc[k]);

if (j != 1)

{

Console.Write(",");

}

}

Console.WriteLine(" ");

Console.WriteLine("----------d--------------");

Console.Write("");

for (int j = cdsize[1]+1, k = 1; j > 0; j--, k++)

{

revd[k] = d[j];

Console.Write(revd[k]);

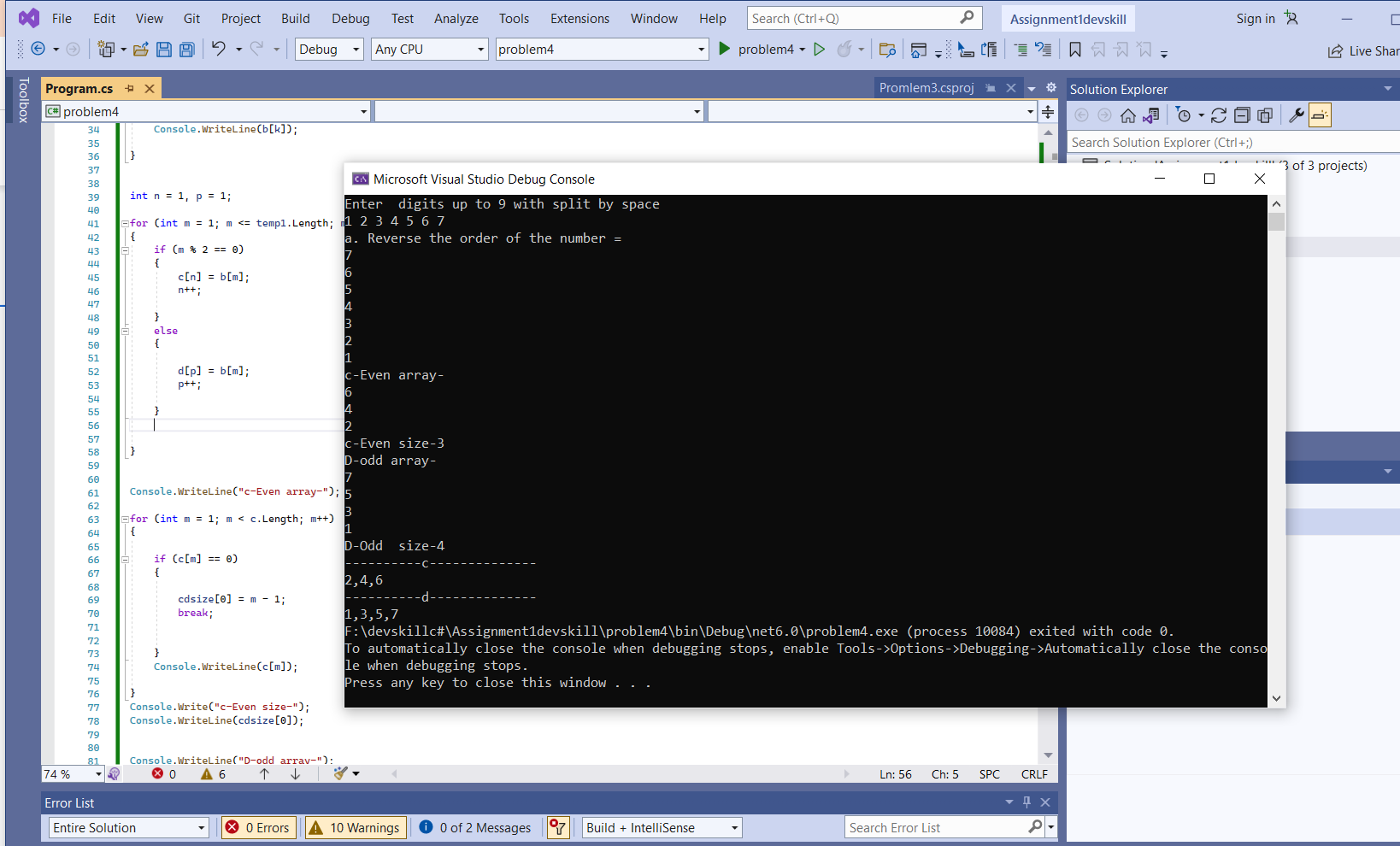
if (j != 1)

{

Console.Write(",");

}

}

****

**Problem-5.**

int[] a = new int[9];

int[] b = new int[3];

int[] c = new int[3];

Console.WriteLine("Enter 3X3 Matrix Elements");

string arrayTemp1 = Convert.ToString(Console.ReadLine());

string[] temp1 = arrayTemp1.Split(' ');

for (int i = 0; i < temp1.Length; i++) {

a[i] = Convert.ToInt32(temp1[i]);

}

string arrayTemp2 = Convert.ToString(Console.ReadLine());

string[] temp2 = arrayTemp2.Split(' ');

for (int i = 0; i < temp2.Length; i++)

{

b[i] = Convert.ToInt32(temp2[i]);

}

string arrayTemp3 = Convert.ToString(Console.ReadLine());

string[] temp3 = arrayTemp3.Split(' ');

for (int i = 0; i < temp3.Length; i++)

{

c[i] = Convert.ToInt32(temp3[i]);

}

for (int m = 0; m < 3; m++) {

a[3+m]= b[m];

a[6 + m] = c[m];

}

int j, k,s1=0,s2=0;

int[,] matx = new int[3, 3];

for(j=0; j<3; j++)

{

for ( k = 0; k < 3; k++) {

if (j == 0)

{ matx[j, k] = a[k]; }

if (j == 1)

{ matx[j, k] = a[k+3]; }

if (j == 2)

{ matx[j, k] = a[k + 6]; }

}

}

for (j = 0; j < 3; j++)

{

for (k = 0; k < 3; k++)

{

if (j == k)

{

s1 = s1 + matx[j, k];

}

}

}

String s = String.Format("1st diagonal value = {0}", s1);

Console.WriteLine(s);

for (j = 0; j < 3; j++)

{

for (k = 2; k >=0; k--)

{

if (j + k == 2)

{ s2 = s2 + matx[j, k];

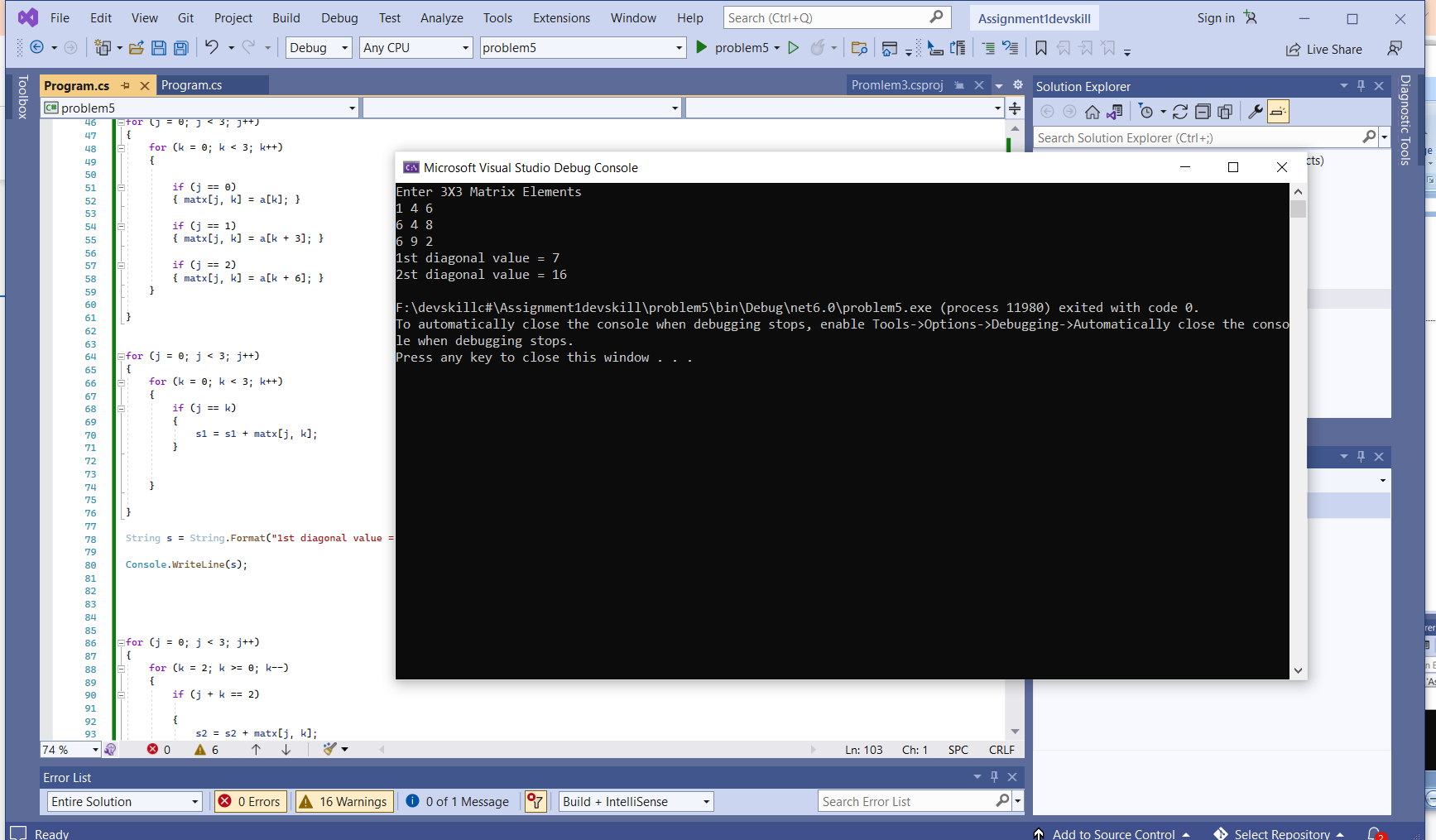
}

}

}

String p = String.Format("2st diagonal value = {0} ", s2);

Console.WriteLine(p);



**Problem-7**

int[] a = new int[9];

int[] b = new int[3];

int[] c = new int[3];

int[] t = new int[3];

int[] n = new int[3];

int[] p = new int[3];

int[] v = new int[9];

Console.WriteLine("Enter 3X3 Matrix Elements");

string arrayTemp1 = Convert.ToString(Console.ReadLine());

string[] temp1 = arrayTemp1.Split(' ');

for (int i = 0; i < temp1.Length; i++)

{

a[i] = Convert.ToInt32(temp1[i]);

}

string arrayTemp2 = Convert.ToString(Console.ReadLine());

string[] temp2 = arrayTemp2.Split(' ');

for (int i = 0; i < temp2.Length; i++)

{

b[i] = Convert.ToInt32(temp2[i]);

}

string arrayTemp3 = Convert.ToString(Console.ReadLine());

string[] temp3 = arrayTemp3.Split(' ');

for (int i = 0; i < temp3.Length; i++)

{

c[i] = Convert.ToInt32(temp3[i]);

}

Console.WriteLine("Enter Rotation Number =");

string numbersLine = Console.ReadLine();

int rotation = Convert.ToInt32(numbersLine)%4;

for (int m = 0; m < 3; m++)

{

a[3 + m] = b[m];

a[6 + m] = c[m];

}

int j, k, s1 = 0, s2 = 0;

int[,] matx = new int[3, 3];

for (j = 0; j < 3; j++)

{

for (k = 0; k < 3; k++)

{

if (j == 0)

{ matx[j, k] = a[k]; }

if (j == 1)

{ matx[j, k] = a[k + 3]; }

if (j == 2)

{ matx[j, k] = a[k + 6]; }

}

}

Console.WriteLine("----Your Input--");

for (j = 0; j < 3; j++)

{

for (k = 0; k < 3; k++)

{

Console.Write(matx[j,k]);

Console.Write(' ');

}

Console.WriteLine("");

}

for (j = 0; j <3; j++ )

{

for (k = 0; k < 3; k++)

{

if (j == 0)

{ t[k] = c[k];

}

if (j == 1)

{ n[k] = b[k]; }

if (j == 2)

{

p[k] = a[k];

}

}

Console.WriteLine("");

}

if (rotation ==1)

{

for (int h = 0, l = 0; h < 3; h++, l = l + 3)

{

v[l] = t[h];

v[l + 1] = n[h];

v[l + 2] = p[h]; }

}

if (rotation == 2)

{

for(int f=0; f < 3;) {

for (int h = 0,l=2; h < 3; h++)

{

v[h] = t[l];

v[l + f+h+1] = n[l];

v[l + f+h+4] = p[l];

l = l - 1;

f=f + 1;

}

}

}

if (rotation == 3)

{

for (int f = 0; f < 3;)

{

for (int h = 0, l = 2; h <9; h = h + 3)

{

v[h] = p[l];

v[l + f + h - 1] = n[l];

v[l + f + h ] = t[l];

l = l - 1;

f = f + 1;

}

}

}

int[,] rot1 = new int[3, 3];

for (j = 0; j < 3; j++)

{

for (k = 0; k < 3; k++)

{

if (j == 0)

{ rot1[j, k] = v[k]; }

if (j == 1)

{ rot1[j, k] = v[k + 3]; }

if (j == 2)

{ rot1[j, k] = v[k + 6]; }

}

}

Console.WriteLine("--Rotation result=-----");

for (j = 0; j < 3; j++)

{

for (k = 0; k < 3; k++)

{

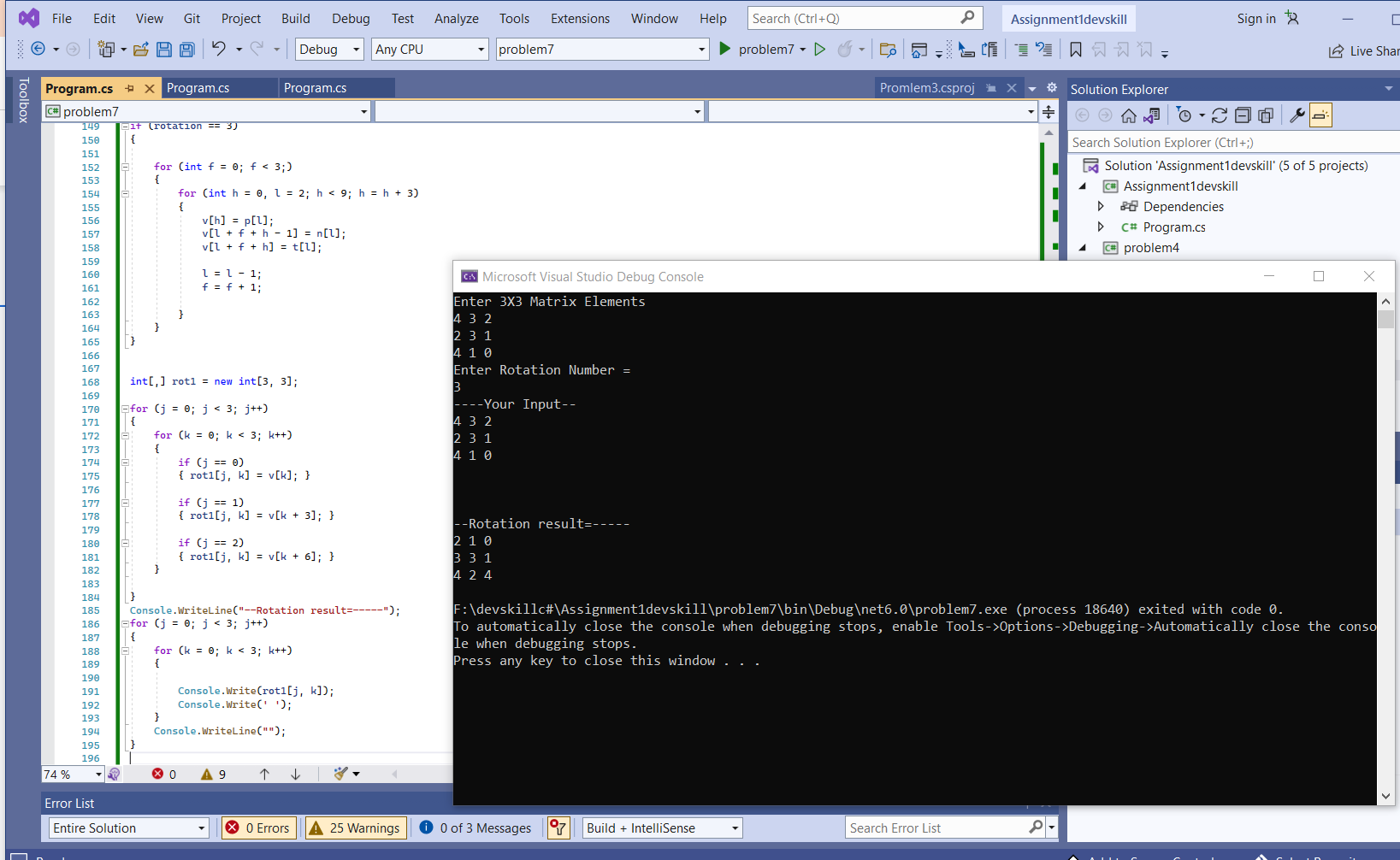
Console.Write(rot1[j, k]);

Console.Write(' ');

}

Console.WriteLine("");

}

****