به نام خدا

برنامه سازی

مرکز تحقیقات معلمان
۱۳۹۴/۰۸/۱۷
کامپیوتر حرفه ای

Number of Operation	Nature of Operation	Variables acted upon	Variables receiving results	Indication of change in the value on any Variable	Statement of Results	Data			Working Variables										Result Variables			
						${}^1V_1$	${}^1V_2$	${}^1V_3$	${}^0V_4$	${}^0V_5$	${}^0V_6$	${}^0V_7$	${}^0V_8$	${}^0V_9$	${}^0V_{10}$	${}^0V_{11}$	${}^0V_{12}$	${}^0V_{13}\ldots$	${}^1V_{21}$	${}^1V_{22}$	${}^1V_{23}$	${}^0V_{24}\ldots$
						○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
						0	0	0	0	0	0	0	0	0	0	0	0	0	B in a dec. fract.	B in a dec. fract.	B in a dec. fract.	0
						0	0	0	0	0	0	0	0	0	0	0	0	0				0
						1	2	4	0	0	0	0	0	0	0	0	0	0				0
						$\boxed{1}$	$\boxed{2}$	$\boxed{n}$	□	□	□	□	□	□	□	□	□	□	$\boxed{B_1}$	$\boxed{B_3}$	$\boxed{B_5}$	$\boxed{B_7}$
1	×	${}^1V_2 \times {}^1V_3$	${}^1V_4,\ {}^1V_5,\ {}^1V_6$	$\left\{\begin{matrix} {}^1V_2 = {}^1V_2 \\ {}^1V_3 = {}^1V_3 \end{matrix}\right\}$	$= 2n$	......	2	n	$2n$	$2n$	$2n$											
2	−	${}^1V_4 - {}^1V_1$	${}^2V_4$	$\left\{\begin{matrix} {}^1V_4 = {}^2V_4 \\ {}^1V_1 = {}^1V_1 \end{matrix}\right\}$	$= 2n - 1$	1	......	......	$2n - 1$													
3	+	${}^1V_5 + {}^1V_1$	${}^2V_5$	$\left\{\begin{matrix} {}^1V_5 = {}^2V_5 \\ {}^1V_1 = {}^1V_1 \end{matrix}\right\}$	$= 2n + 1$	1	......	......	......	$2n + 1$												
4	÷	${}^2V_5 \div {}^2V_4$	${}^1V_{11}$	$\left\{\begin{matrix} {}^2V_5 = {}^0V_5 \\ {}^2V_4 = {}^0V_4 \end{matrix}\right\}$	$= \frac{2n-1}{2n+1}$	......	......	......	0	0	......	......	......	......	......	$\frac{2n-1}{2n+1}$						
5	÷	${}^1V_{11} \div {}^1V_2$	${}^2V_{11}$	$\left\{\begin{matrix} {}^1V_{11} = {}^2V_{11} \\ {}^1V_2 = {}^1V_2 \end{matrix}\right\}$	$= \frac{1}{2} \cdot \frac{2n-1}{2n+1}$	......	2	......	......	......	......	......	......	......	......	$\frac{1}{2} \cdot \frac{2n-1}{2n+1}$						
6	−	${}^0V_{13} - {}^2V_{11}$	${}^1V_{13}$	$\left\{\begin{matrix} {}^2V_{11} = {}^0V_{11} \\ {}^0V_{13} = {}^1V_{13} \end{matrix}\right\}$	$= -\frac{1}{2} \cdot \frac{2n-1}{2n+1} = A_0$	......	......	......	......	......	......	......	......	......	......	0	....	$= -\frac{1}{2} \cdot \frac{2n-1}{2n+1} = A_0$				
7	−	${}^1V_3 - {}^1V_1$	${}^1V_{10}$	$\left\{\begin{matrix} {}^1V_3 = {}^1V_3 \\ {}^1V_1 = {}^1V_1 \end{matrix}\right\}$	$= n - 1 (= 3)$	1	......	n	......	......	......	......	......	......	$n - 1$							
8	+	${}^1V_2 + {}^0V_7$	${}^1V_7$	$\left\{\begin{matrix} {}^1V_2 = {}^1V_2 \\ {}^0V_7 = {}^1V_7 \end{matrix}\right\}$	$= 2 + 0 = 2$	......	2	......	......	......	......	2										
9	÷	${}^1V_6 \div {}^1V_7$	${}^3V_{11}$	$\left\{\begin{matrix} {}^1V_6 = {}^1V_6 \\ {}^0V_{11} = {}^3V_{11} \end{matrix}\right\}$	$= \frac{2n}{2} = A_1$	......	......	......	......	......	$2n$	2	......	......	......	$\frac{2n}{2} = A_1$						
10	×	${}^1V_{21} \times {}^3V_{11}$	${}^1V_{12}$	$\left\{\begin{matrix} {}^1V_{21} = {}^1V_{21} \\ {}^3V_{11} = {}^3V_{11} \end{matrix}\right\}$	$= B_1 \cdot \frac{2n}{2} = B_1A_1$	......	......	......	......	......	......	......	......	......	......	$\frac{2n}{2} = A_1$	$B_1 \cdot \frac{2n}{2} = B_1A_1$	....	B_1			
11	+	${}^1V_{12} + {}^1V_{13}$	${}^2V_{13}$	$\left\{\begin{matrix} {}^1V_{12} = {}^0V_{12} \\ {}^1V_{13} = {}^2V_{13} \end{matrix}\right\}$	$= -\frac{1}{2} \cdot \frac{2n-1}{2n+1} + B_1 \cdot \frac{2n}{2}$	......	......	......	......	......	......	......	......	......	......	....	0	$\left\{-\frac{1}{2} \cdot \frac{2n-1}{2n+1} + B_1 \cdot \frac{2n}{2}\right\}$				
12	−	${}^1V_{10} - {}^1V_1$	${}^2V_{10}$	$\left\{\begin{matrix} {}^1V_{10} = {}^2V_{10} \\ {}^1V_1 = {}^1V_1 \end{matrix}\right\}$	$= n - 2 (= 2)$	1	......	......	......	......	......	......	......	......	$n - 2$							
13	−	${}^1V_6 - {}^1V_1$	${}^2V_6$	$\left\{\begin{matrix} {}^1V_6 = {}^2V_6 \\ {}^1V_1 = {}^1V_1 \end{matrix}\right\}$	$= 2n - 1$	1	......	......	......	......	$2n - 1$											
14	+	${}^1V_1 + {}^1V_7$	${}^2V_7$	$\left\{\begin{matrix} {}^1V_1 = {}^1V_1 \\ {}^1V_7 = {}^2V_7 \end{matrix}\right\}$	$= 2 + 1 = 3$	1	......	......	......	......	......	3										
15	÷	${}^2V_6 \div {}^2V_7$	${}^1V_8$	$\left\{\begin{matrix} {}^2V_6 = {}^2V_6 \\ {}^2V_7 = {}^2V_7 \end{matrix}\right\}$	$= \frac{2n-1}{3}$	......	......	......	......	......	$2n - 1$	3	$\frac{2n-1}{3}$									
16	×	${}^1V_8 \times {}^3V_{11}$	${}^4V_{11}$	$\left\{\begin{matrix} {}^1V_8 = {}^0V_8 \\ {}^3V_{11} = {}^4V_{11} \end{matrix}\right\}$	$= \frac{2n}{2} \cdot \frac{2n-1}{3}$	......	......	......	......	......	......	......	0	......	......	$\frac{2n}{2} \cdot \frac{2n-1}{3}$						
17	−	${}^2V_6 - {}^1V_1$	${}^3V_6$	$\left\{\begin{matrix} {}^2V_6 = {}^3V_6 \\ {}^1V_1 = {}^1V_1 \end{matrix}\right\}$	$= 2n - 2$	1	......	......	......	......	$2n - 2$											
18	+	${}^1V_1 + {}^2V_7$	${}^3V_7$	$\left\{\begin{matrix} {}^2V_7 = {}^3V_7 \\ {}^1V_1 = {}^1V_1 \end{matrix}\right\}$	$= 3 + 1 = 4$	1	......	......	......	......	......	4										
19	÷	${}^3V_6 \div {}^3V_7$	${}^1V_9$	$\left\{\begin{matrix} {}^3V_6 = {}^3V_6 \\ {}^3V_7 = {}^3V_7 \end{matrix}\right\}$	$= \frac{2n-2}{4}$	......	......	......	......	......	$2n - 2$	4	......	$\frac{2n-2}{4}$	......							
20	×	${}^1V_9 \times {}^4V_{11}$	${}^5V_{11}$	$\left\{\begin{matrix} {}^1V_9 = {}^0V_9 \\ {}^4V_{11} = {}^5V_{11} \end{matrix}\right\}$	$= \frac{2n}{2} \cdot \frac{2n-1}{3} \cdot \frac{2n-2}{4} = A_3$	......	......	......	......	......	......	......	......	0		$\left\{\frac{2n}{2} \cdot \frac{2n-1}{3} \cdot \frac{2n-2}{4}\right\} = A_3$						
21	×	${}^1V_{22} \times {}^5V_{11}$	${}^0V_{12}$	$\left\{\begin{matrix} {}^1V_{22} = {}^1V_{22} \\ {}^0V_{12} = {}^2V_{12} \end{matrix}\right\}$	$= B_3 \cdot \frac{2n}{2} \cdot \frac{2n-1}{3} \cdot \frac{2n-2}{4} = B_3A_3$	......	......	......	......	......	......	......	......	......	......	0	B_3A_3	....	......	B_3		
22	+	${}^2V_{12} + {}^2V_{13}$	${}^3V_{13}$	$\left\{\begin{matrix} {}^2V_{12} = {}^0V_{12} \\ {}^2V_{13} = {}^3V_{13} \end{matrix}\right\}$	$= A_0 + B_1A_1 + B_3A_3$	......	......	......	......	......	......	......	......	......	......	....	0	$\{A_0 + B_1A_1 + B_3A_3\}$				
23	−	${}^2V_{10} - {}^1V_1$	${}^3V_{10}$	$\left\{\begin{matrix} {}^2V_{10} = {}^3V_{10} \\ {}^1V_1 = {}^1V_1 \end{matrix}\right\}$	$= n - 3 (= 1)$	1	......	......	......	......	......	......	......	......	$n - 3$							
Here follows a repetition of Operations thirteen to twenty-three																						
24	+	${}^4V_{13} + {}^0V_{24}$	${}^1V_{24}$	$\left\{\begin{matrix} {}^4V_{13} = {}^0V_{13} \\ {}^0V_{24} = {}^1V_{24} \end{matrix}\right\}$	$= B_7$	......	......	......	......	......	......	......	......	......	......	....	....	....	......	......	......	B_7
25	+	${}^1V_1 + {}^1V_3$	${}^1V_3$	$\left\{\begin{matrix} \left.\begin{matrix} {}^1V_1 = {}^1V_1 \\ {}^1V_3 = {}^1V_3 \end{matrix}\right\} \\ {}^5V_6 = {}^0V_6 \\ {}^5V_7 = {}^0V_7 \end{matrix}\right.$	$= n + 1 = 4 + 1 = 5$ by a Variable-card. by a Variable-card.	1	......	$n + 1$	......	......	0	0										

Semantic

مثال: ساخت مثلث

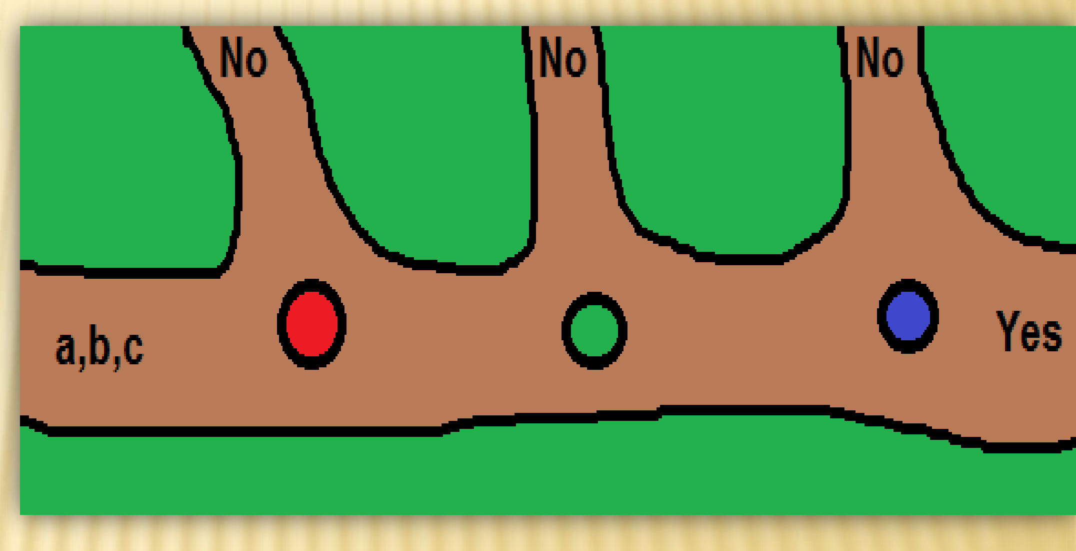
راه اول:

```
 float a = 3, b = 4, c = 5;
 int t=0;
 if (a + b > c) ++t;
 if (a + c > b) ++t;
 if (b + c > a) ++t;
 if (t == 3) Console.WriteLine(true);
else Console.WriteLine(false);
```

ساخت مثلث

راه دوم

```
float a = 3, b = 4, c = 5;
 if (a + b <= c)
      Console.WriteLine(false);
 else if (a + c <= b)
      Console.WriteLine(false);
 else if (b + c <= a)
      Console.WriteLine(false);
 else
       Console.WriteLine(true);
```



ساخت مثلث

راه سوم:

First:

```
float a = 3, b = 4, c = 5;
 if (a + b <= c || a + c <= b || b + c<= a)
     Console.WriteLine(false );
else
      Console.WriteLine(true );
```

Second:

```
 float a = 3, b = 4, c = 5;
  if (a + b > c && a + c > b && b + c > a)
      Console.WriteLine(true);
  else
       Console.WriteLine(false);
```

!(P | | Q)=!P && !Q

ساخت مثلث

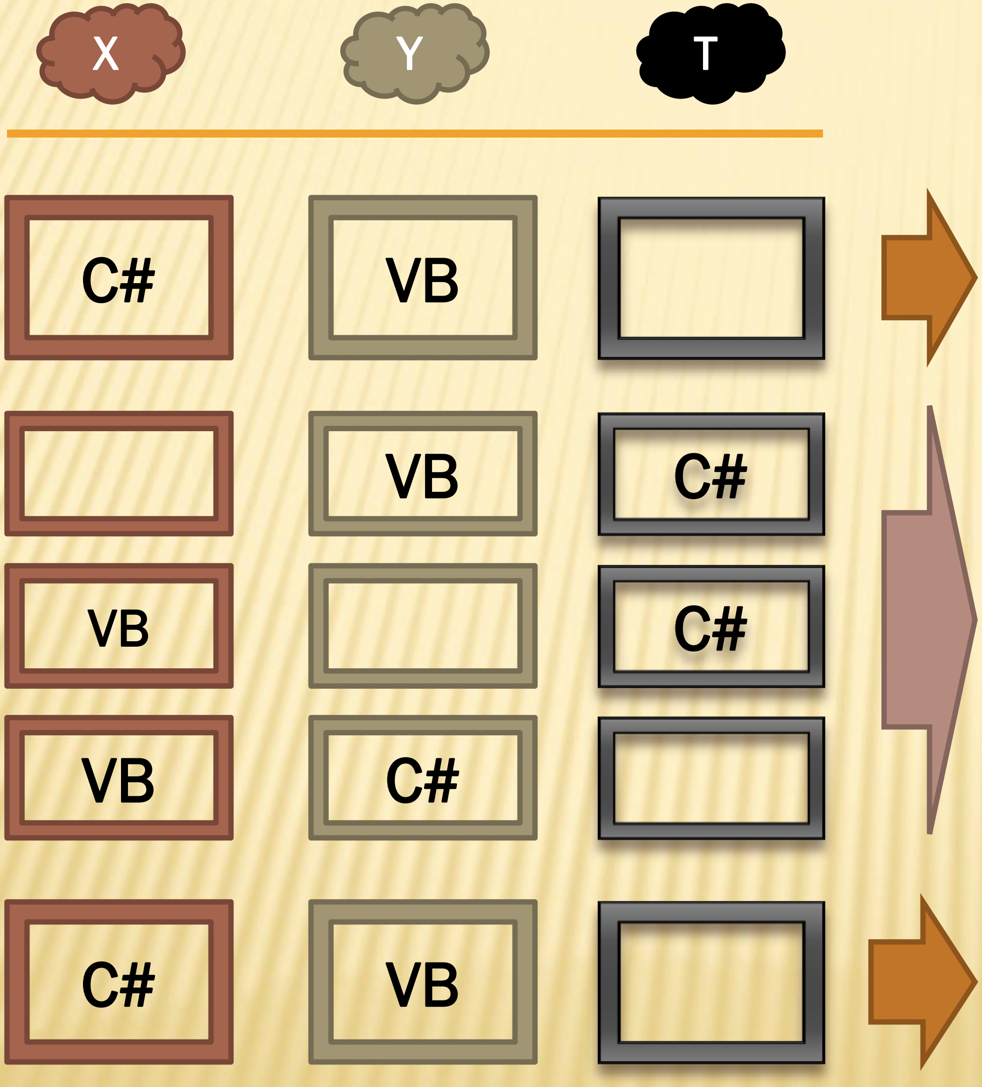
راه چهارم:

```
float a = 3, b = 4, c = 5;
Console.WriteLine  (a + b > c && a + c > b && b + c > a?true:false );
```

```
float a = 3, b = 4, c = 5;
 Console.WriteLine  (a + b > c && a + c > b && b + c > a);
```

مثال ۲:جابجایی داده

راه اول



X	Y	T
C#	VB	
	VB	C#
VB		C#
VB	C#	
C#	VB	

```
string X = "VB", Y = "C#", T;

T = X;

X = Y;

Y = T;

Console.WriteLine("{0},{1}",X,Y);
```

تعیین الگوی به خاطر سپاری

TXXYYT

XAABBX

X=A;
A=B;
B=X;

جابجایی داده

راه دوم

```
 int X = 2, Y = 5, T;
X = X+Y;
Y = X-Y;
X = X-Y;
Console.WriteLine("{0},{1}",X,Y);
```

```
 int X = 2, Y = 5, T;
X += Y;
Y = X-Y;
X -= Y;
Console.WriteLine("{0},{1}",X,Y);
```

جابجایی داده

راه سوم

```
int X = 2, Y = 5, T;
X = X*Y;
Y = X/Y;
X = X/Y;
Console.WriteLine("{0},{1}",X,Y);
```

```
int X = 2, Y = 5, T;
X *= Y;
Y = X/Y;
X /= Y;
Console.WriteLine("{0},{1}",X,Y);
```

جابجایی داده

راه چهارم

```
 int X = 2, Y = 5, T;
X = X^Y;
Y = X^Y;
X = X^Y;
Console.WriteLine("{0},{1}",X,Y);
```

```
 int X = 2, Y = 5, T;
X ^= Y;
Y ^= X;
X ^= Y;
Console.WriteLine("{0},{1}",X,Y);
```

جابجایی داده

راه پنجم:

```
#include "stdafx.h"


int _tmain(int argc, _TCHAR* argv[])
{
int x=4,y=7;
x^=y^=x^=y;
printf("\n%d,%d\n",x,y);
return 0;
}
```

جابجایی داده(زیر برنامه)

راه ششم:

```
     static void swap( ref int x,  ref int y)
    {
        int t;
        t = x;
        x = y;
        y = t;

    }
static void Main(string[] args)
    {
        float X = 2, Y = 5;
        swap( ref X,  ref Y);
        Console.WriteLine("{0},{1}",X,Y);

    }
```

جابجایی داده(OVERLOAD)

راه ششم:

```
     static void swap( ref int x,  ref int y)
    {
        int t;
        t = x;
        x = y;
        y = t;

    }
static void swap( ref float x,  ref float y)
    {
        float t;
        t = x;
        x = y;
        y = t;
     }
```

جابجایی داده(UNSAFE)

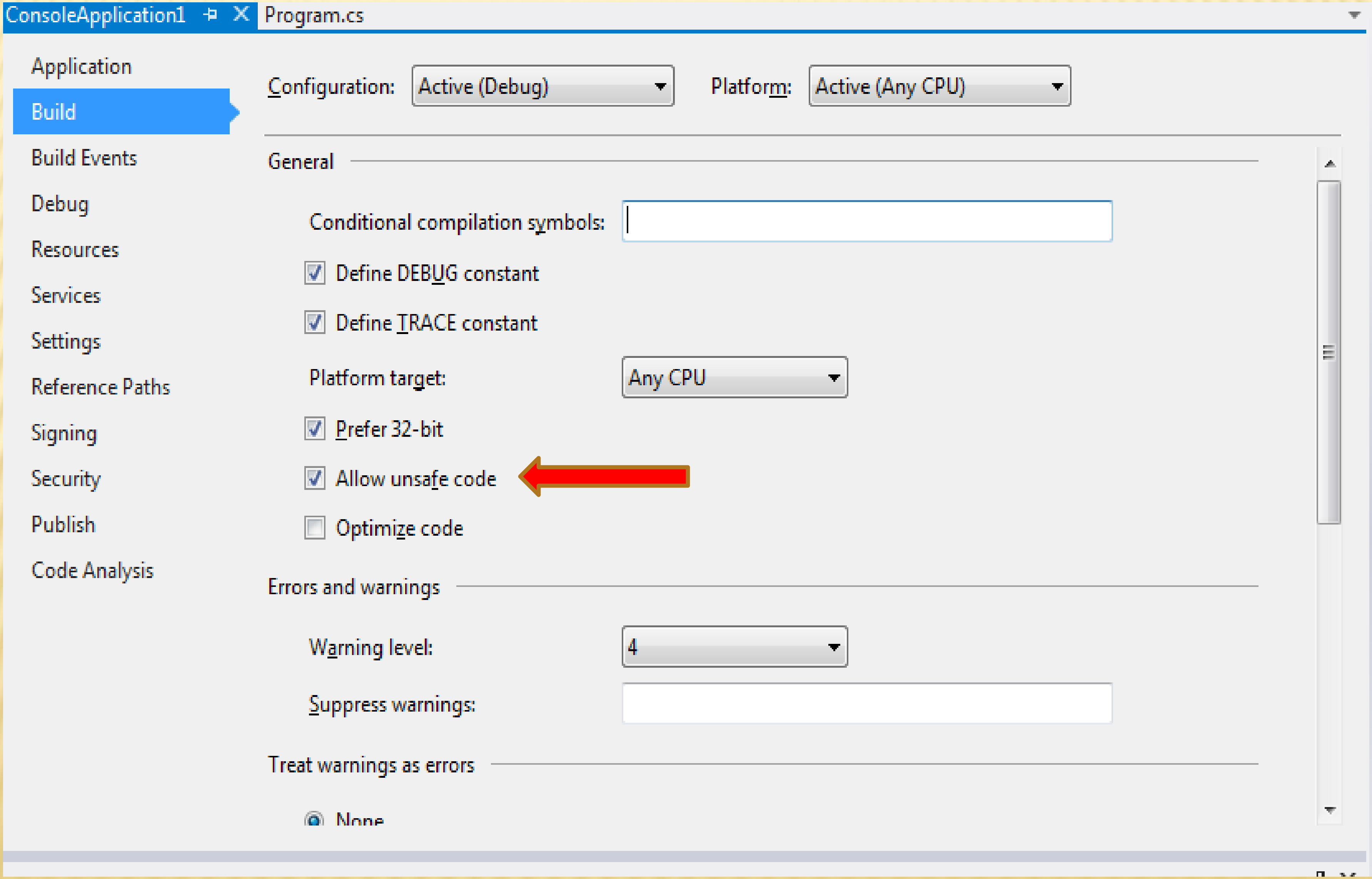
راه هفتم:

```
unsafe static void swap(  float* x,   float* y)
 {
     float t;
     t = *x;
     *x = *y;
     *y = t;

 }
unsafe static void Main(string[] args)
 {
     float X = 2, Y = 5;
     swap( &X,  &Y);
     Console.WriteLine("{0},{1}",X,Y);

 }
```

جابجایی داده(UNSAFE)



Configuration: Active (Debug)

Platform: Active (Any CPU)

General

Conditional compilation symbols:

☑ Define DEBUG constant

☑ Define TRACE constant

Platform target: Any CPU

☑ Prefer 32-bit

☑ Allow unsafe code

□ Optimize code

Errors and warnings

Warning level: 4

Suppress warnings:

Treat warnings as errors

● None

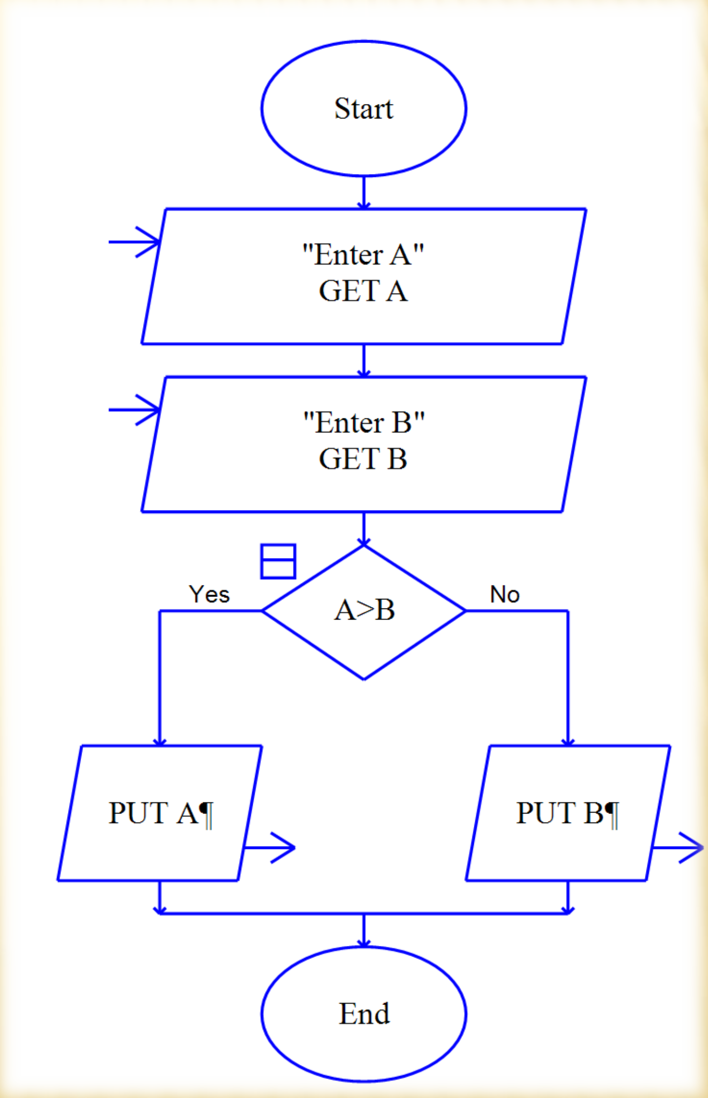
جابجایی داده(GENERIC)

راه هشتم:

```
static void swap<T>(ref T x, ref T y)
        {
            T t = x;
            x = y;
            y = t;
        }
        static void Main(string[] args)
        {
            double X = 2, Y = 5;
            swap(ref X, ref Y);
            Console.WriteLine("{0},{1}",X,Y);
        }
```

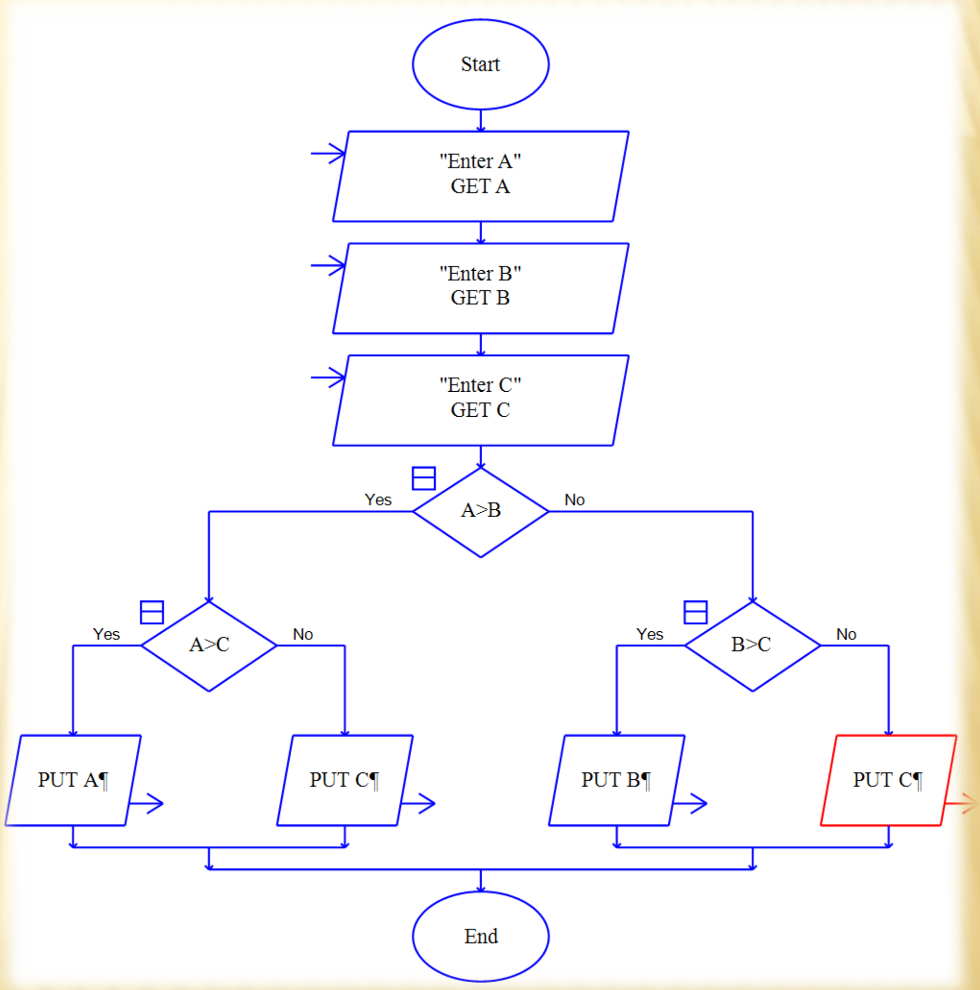
بیشترین دو عدد

```
 int a, b;
a = int.Parse(Console.ReadLine());
b = int.Parse(Console.ReadLine());
if (a>b)
    Console.WriteLine(a);
else
    Console.WriteLine( b);
```



بیشترین سه عدد

```
 int a, b,c;
 a = int.Parse(Console.ReadLine());
 b = int.Parse(Console.ReadLine());
 c = int.Parse(Console.ReadLine());
if (a>b)
 if (a>c)
   Console.WriteLine(a);
 else
   Console.WriteLine(c);
else
 if (b>c)
   Console.WriteLine(b);
  else
   Console.WriteLine(c);
```

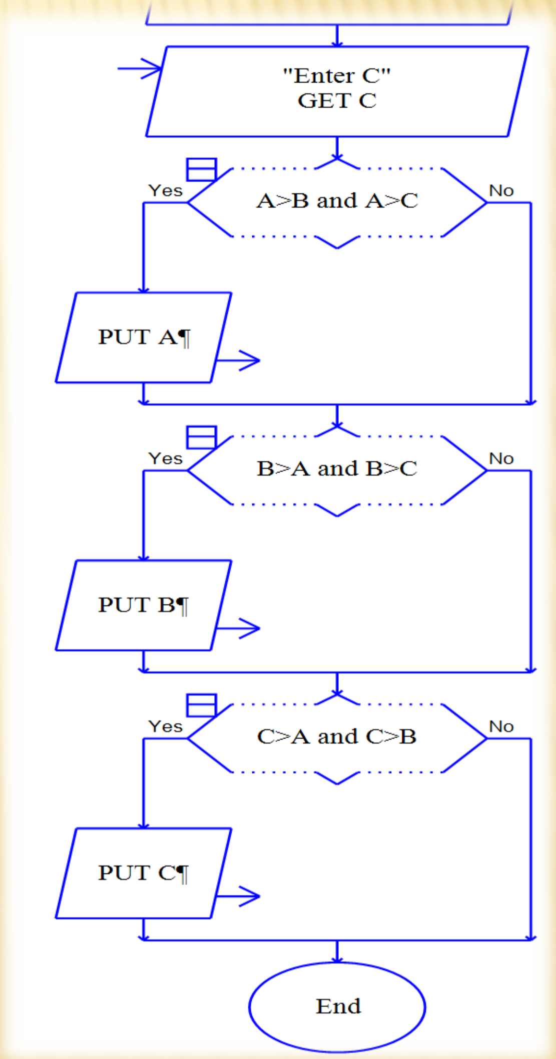


بیشترین چهار عدد

?

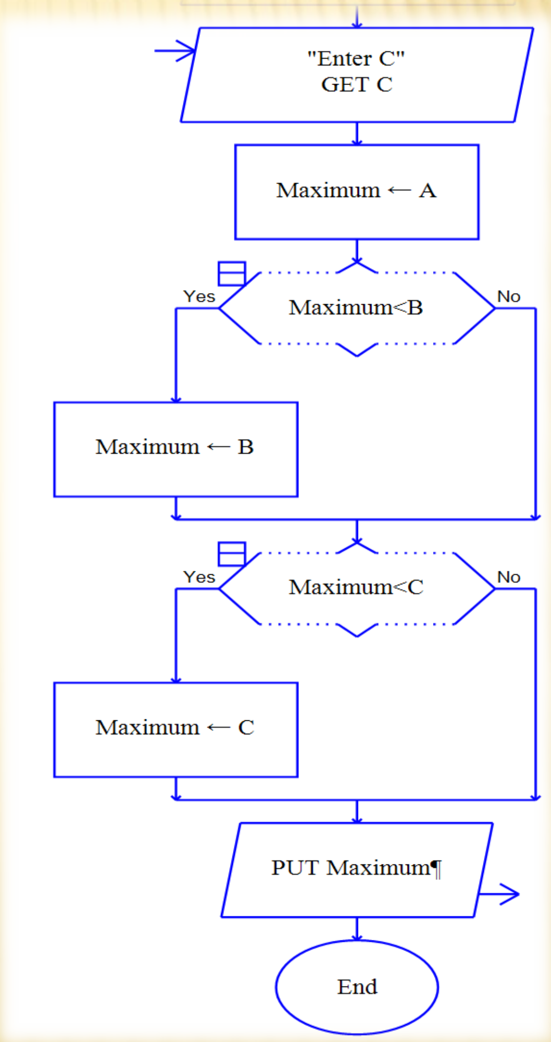
بیشترین سه عدد

```
int a, b,c;
a = int.Parse(Console.ReadLine());
b = int.Parse(Console.ReadLine());
c = int.Parse(Console.ReadLine());
if (a > b && a > c)    Console.WriteLine(a);
if (b > a && b > c)     Console.WriteLine(b);
if (c > a && c > b)    Console.WriteLine(c);
```



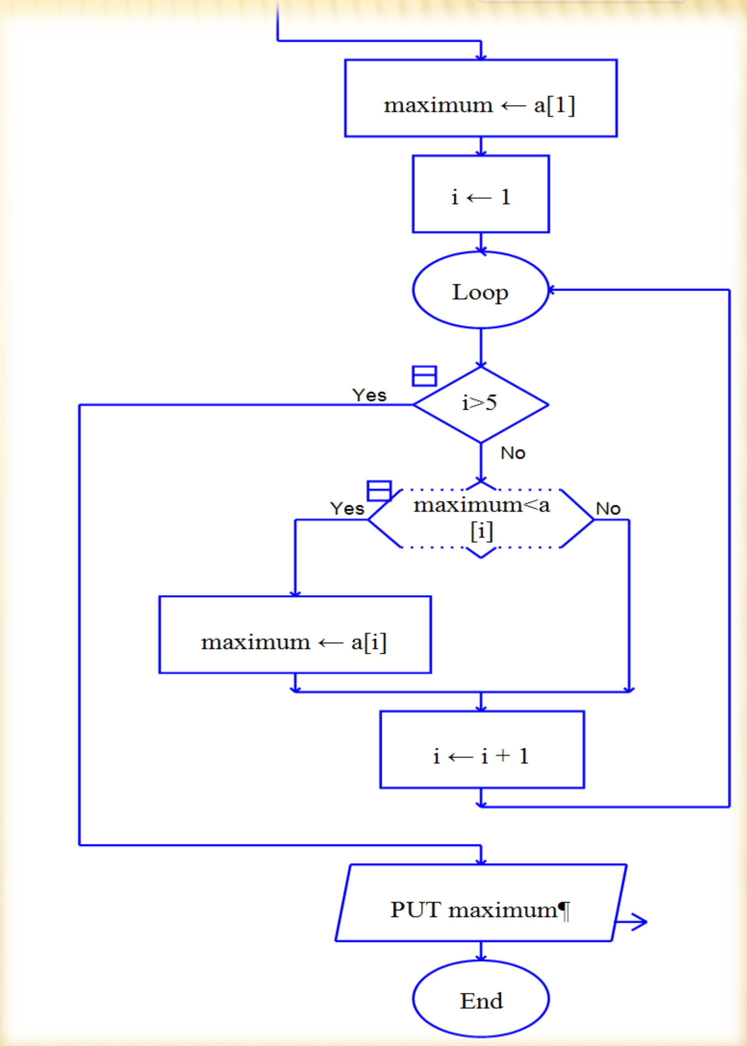
بیشترین سه عدد

```
int a, b,c,maximum;
a = int.Parse(Console.ReadLine());
b = int.Parse(Console.ReadLine());
c = int.Parse(Console.ReadLine());
maximum = a;
if (maximum < b) maximum = b;
if (maximum<c) maximum=c;
Console.WriteLine(maximum );
```



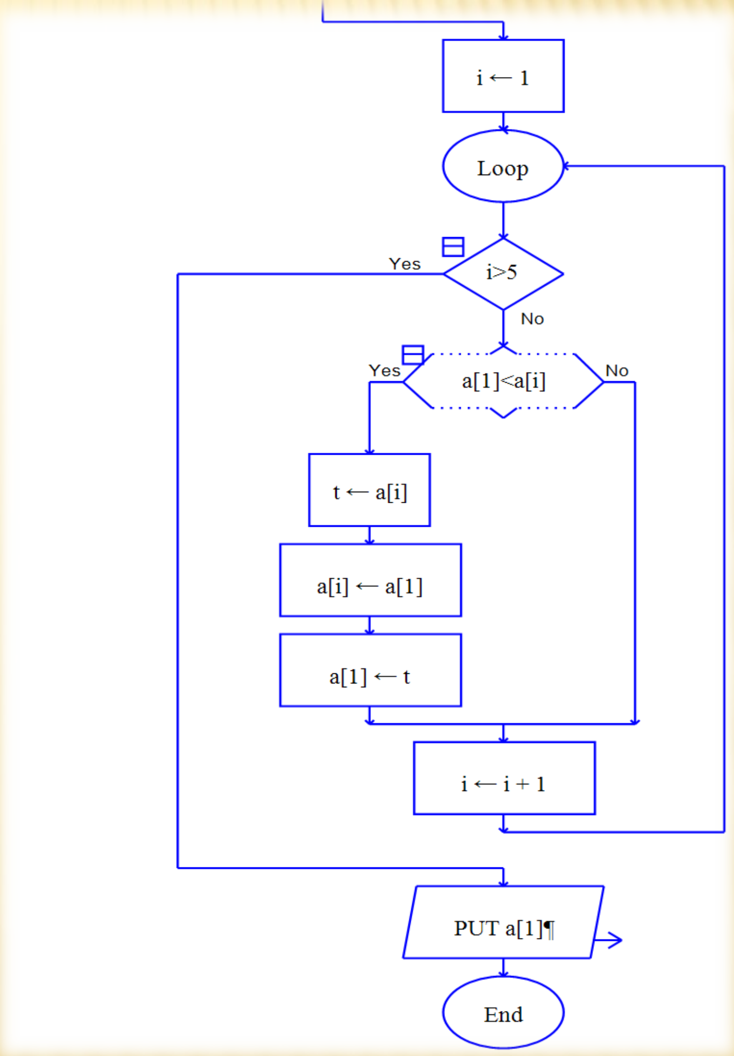
بیشترین پنج عدد

```
//defenition and input data
const int SIZE = 5;
int[] number=new int[SIZE];
for (int i = 0; i < SIZE; i++)
  number[i]=int.Parse(Console.ReadLine());
 //process
 int maximum = number[0];
 for (int i = 0; i < SIZE ; i++)
         if (maximum<number [i])
               maximum = number [i];
 //output
 Console.WriteLine(maximum );
```



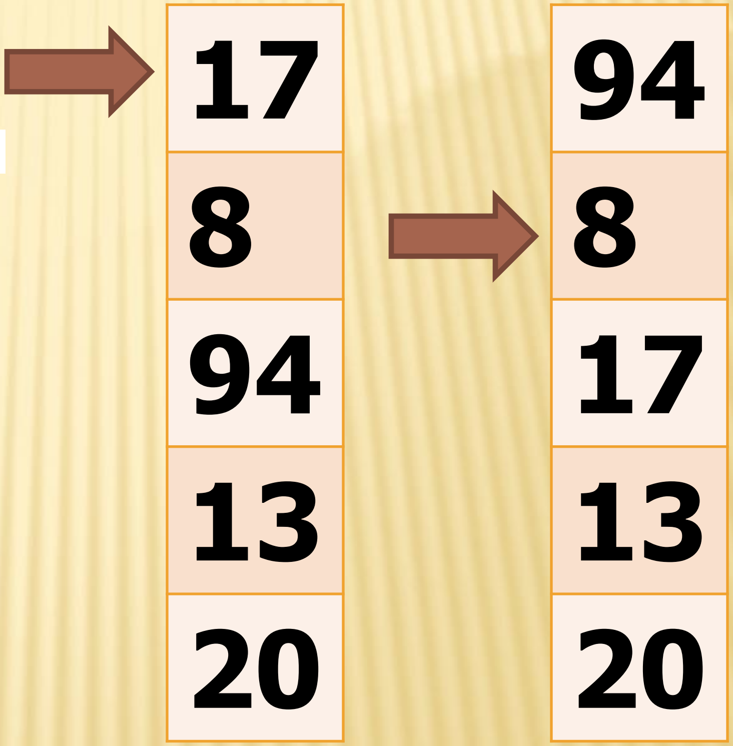
بیشترین پنج عدد

```
const int SIZE = 5;
int[] number=new int[SIZE];
for (int i = 0; i < SIZE; i++)
    number[i]=int.Parse(Console.ReadLine());
//process
for (int i = 0; i < SIZE; i++)
    if (number[0] < number[i])
       {
         int temp = number[i];
         number[i] = number[0];
         number[0] = temp;
        }
//output
Console.WriteLine(number[0]);
```



SELECTION SORTمرتب سازی

```
//defenition and input data
const int SIZE = 5;
int[] number=new int[SIZE];
for (int i = 0; i < SIZE; i++)
    number[i]=int.Parse(Console.ReadLine());
//process
for (int j = 0; j < SIZE ; j++)
{
    for (int i = j; i < SIZE; i++)
        if (number[j] > number[i])
        {
            int temp = number[i];
            number[i] = number[j];
            number[j] = temp;
        }
}
//output
foreach (int item in number )
{
    Console.WriteLine(item );
}
```



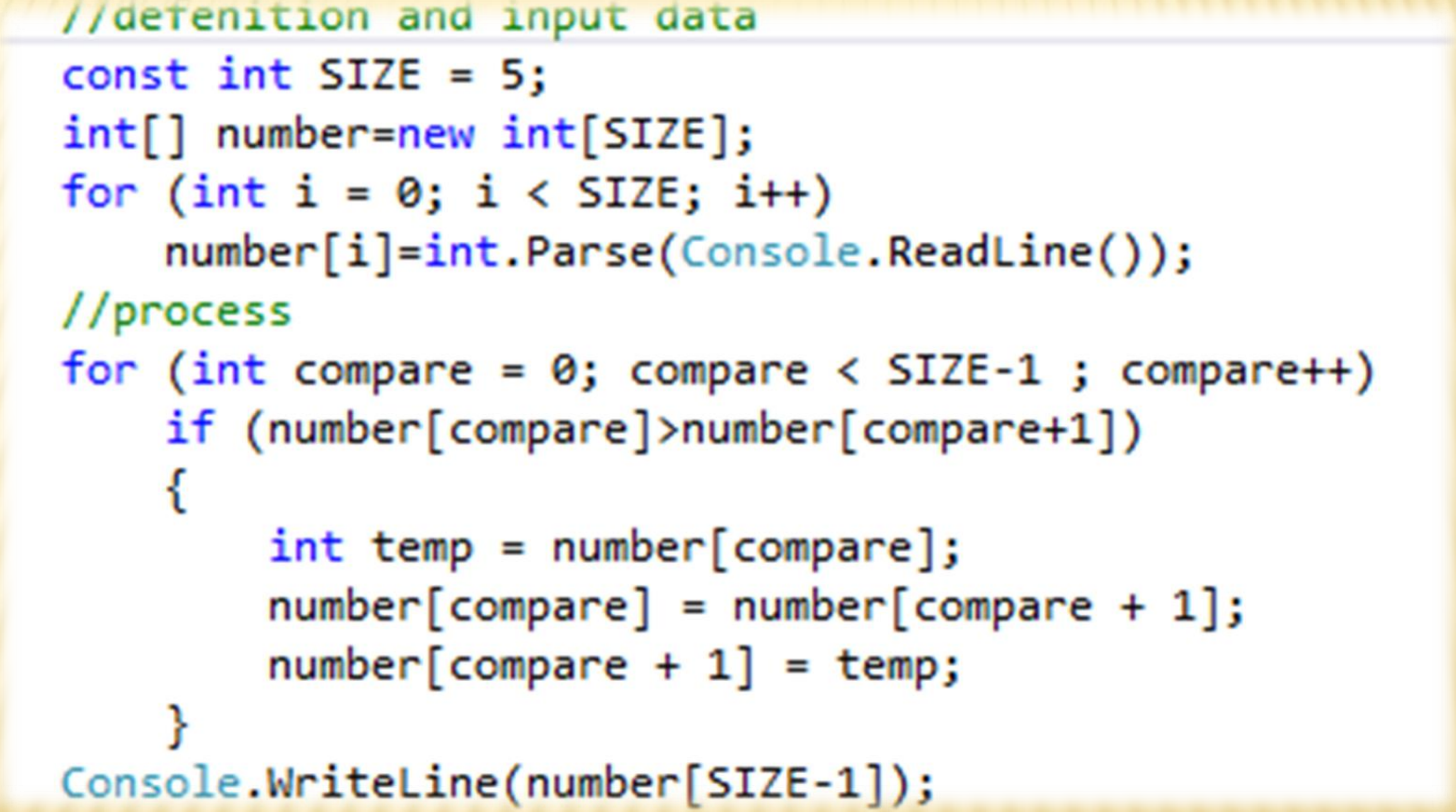
17	94
8	8
94	17
13	13
20	20

مرتب سازی حبابی(BUBBLE SORT)

بیشترین پنج عدد(BUBBLE)

	0	1	2	3	4
⇨	17	8	94	13	20
	8	17	94	13	20
⇨	8	17	94	13	20
⇨	8	17	13	94	20
	8	17	13	20	94

بیشترین پنج عدد(BUBBLE)



```
//defenition and input data
const int SIZE = 5;
int[] number=new int[SIZE];
for (int i = 0; i < SIZE; i++)
    number[i]=int.Parse(Console.ReadLine());
//process
for (int compare = 0; compare < SIZE-1 ; compare++)
    if (number[compare]>number[compare+1])
    {
        int temp = number[compare];
        number[compare] = number[compare + 1];
        number[compare + 1] = temp;
    }
Console.WriteLine(number[SIZE-1]);
```

بیشترین پنج عدد(BUBBLE)

0	1	2	3	4
8	17	13	20	94
8	17	13	20	94
8	13	17	20	94
8	13	17	20	94
8	13	17	20	94

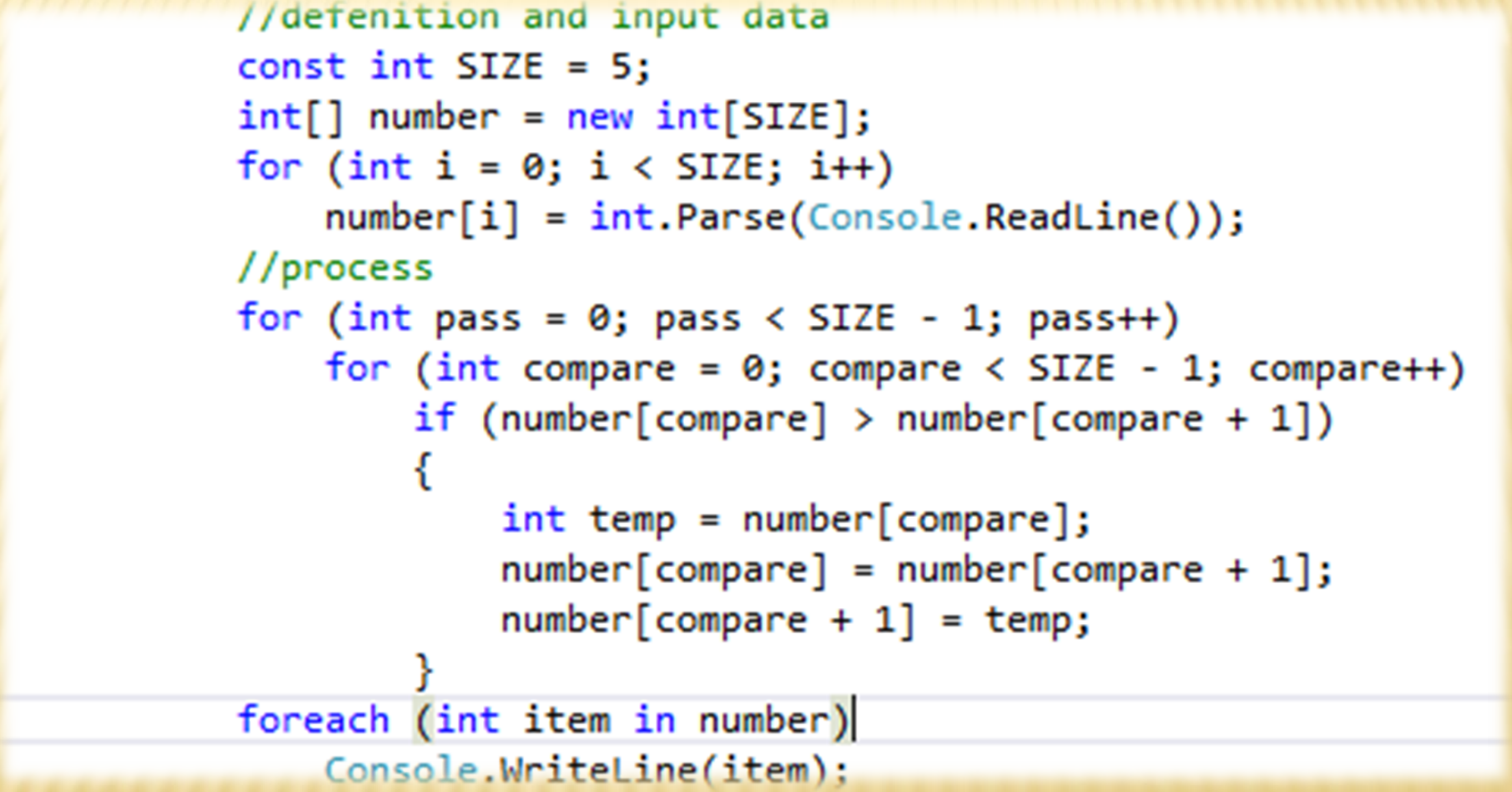
بیشترین پنج عدد(BUBBLE)

0	1	2	3	4
8	13	17	20	94
8	13	17	20	94
8	13	17	20	94
8	13	17	20	94
8	13	17	20	94

بیشترین پنج عدد(BUBBLE)

0	1	2	3	4
8	13	17	20	94
8	13	17	20	94
8	13	17	20	94
8	13	17	20	94
8	13	17	20	94

بیشترین پنج عدد(BUBBLE)



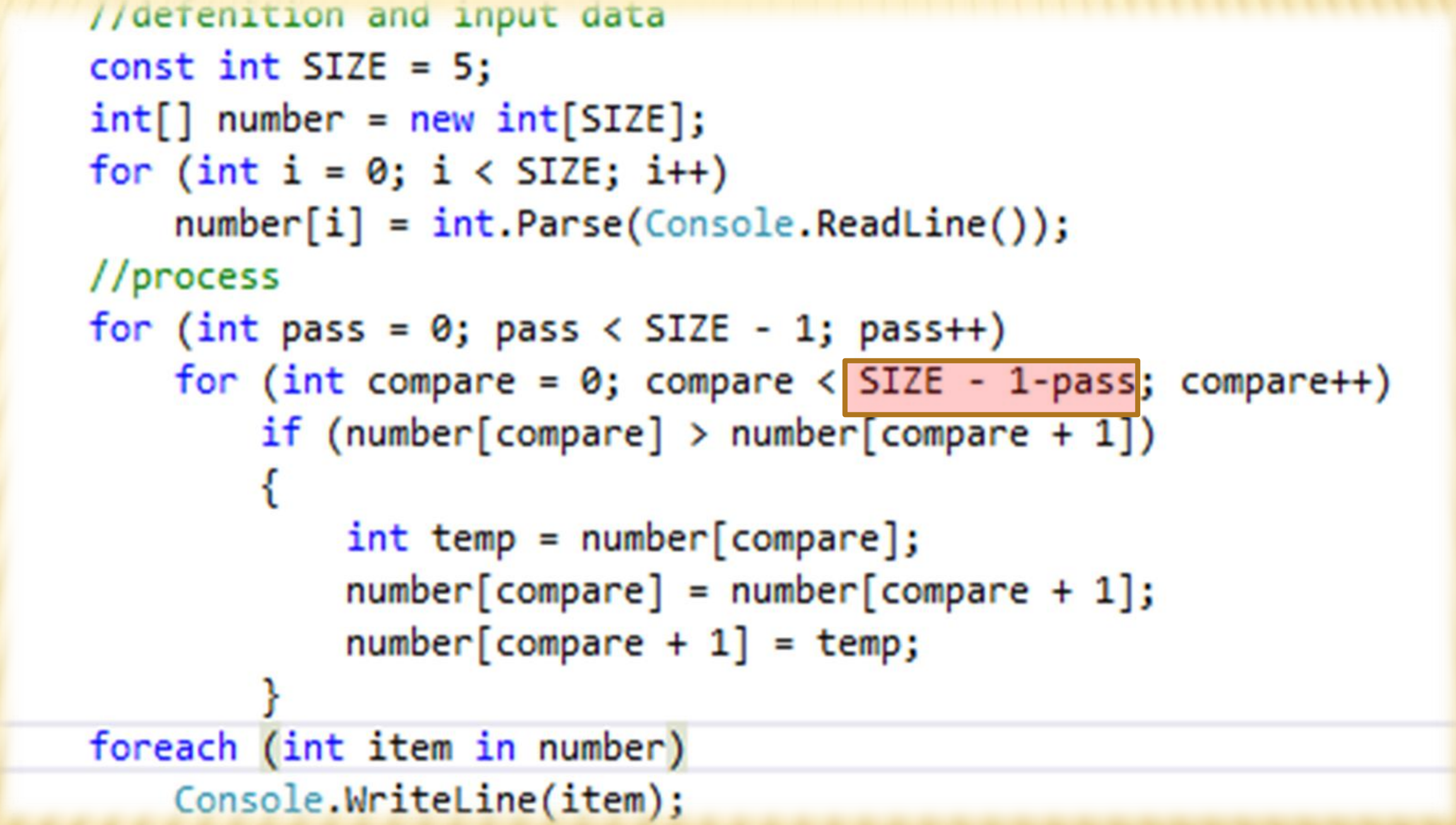
```
//defenition and input data
const int SIZE = 5;
int[] number = new int[SIZE];
for (int i = 0; i < SIZE; i++)
    number[i] = int.Parse(Console.ReadLine());
//process
for (int pass = 0; pass < SIZE - 1; pass++)
    for (int compare = 0; compare < SIZE - 1; compare++)
        if (number[compare] > number[compare + 1])
        {
            int temp = number[compare];
            number[compare] = number[compare + 1];
            number[compare + 1] = temp;
        }
foreach (int item in number)
    Console.WriteLine(item);
```

بیشترین پنج عدد(BUBBLE)

Pass	compares
0	SIZE-1
1	SIZE-1-1
2	SIZE-1-2
SIZE-2	SIZE-1-(SIZE-2)=1
SIZE-1	SIZE-1-(SIZE-1)=0
pass	SIZE-1-pass

Pass	compares
0	SIZE-1
1	SIZE-2
2	SIZE-3
SIZE-2	1
SIZE-1	0

بیشترین پنج عدد(BUBBLE)



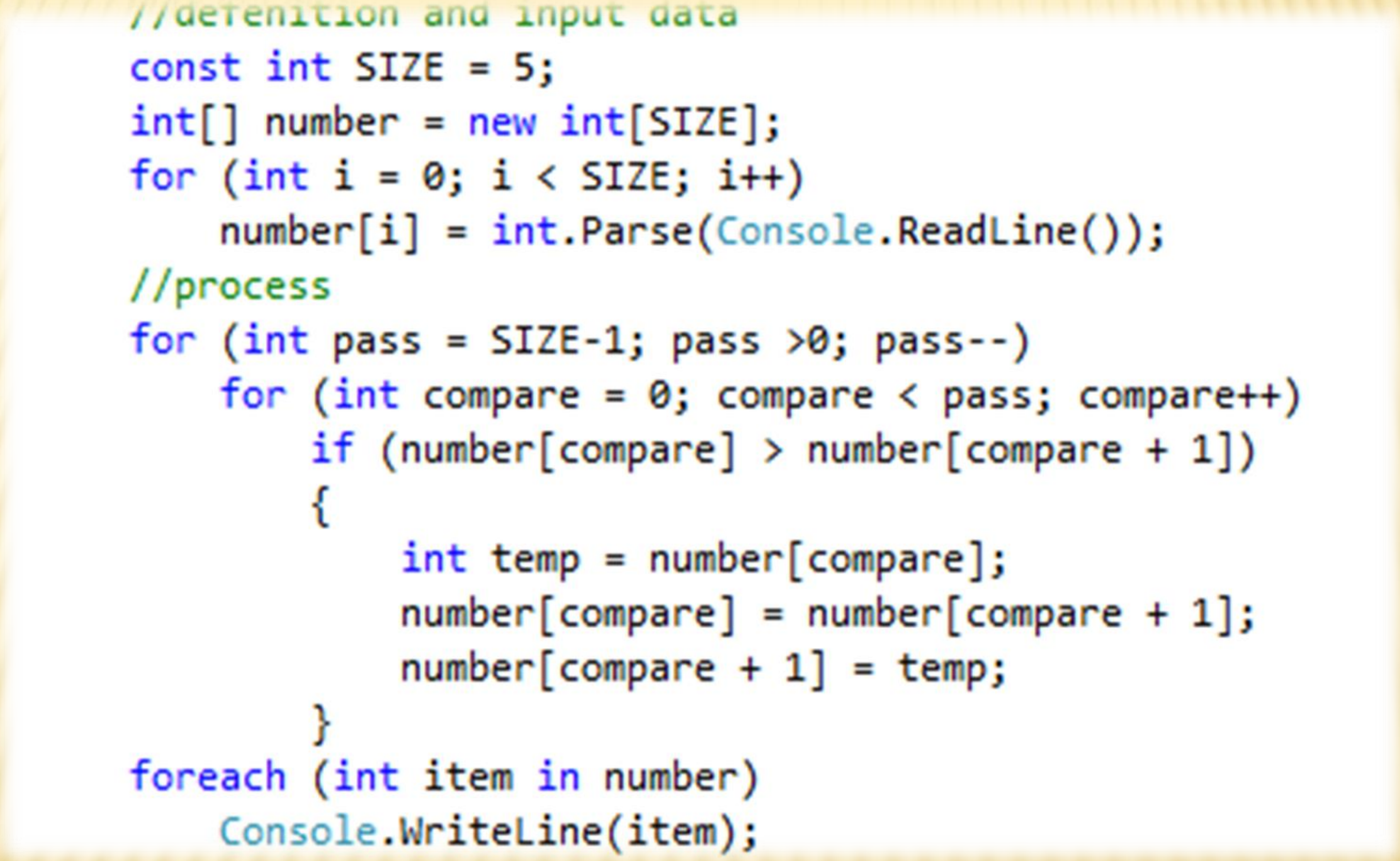
```
//defenition and input data
const int SIZE = 5;
int[] number = new int[SIZE];
for (int i = 0; i < SIZE; i++)
    number[i] = int.Parse(Console.ReadLine());
//process
for (int pass = 0; pass < SIZE - 1; pass++)
    for (int compare = 0; compare < SIZE - 1-pass; compare++)
        if (number[compare] > number[compare + 1])
        {
            int temp = number[compare];
            number[compare] = number[compare + 1];
            number[compare + 1] = temp;
        }
foreach (int item in number)
    Console.WriteLine(item);
```

بیشترین پنج عدد(BUBBLE)

Pass	compares
0	SIZE-1
1	SIZE-2
2	SIZE-3
SIZE-2	1
SIZE-1	0

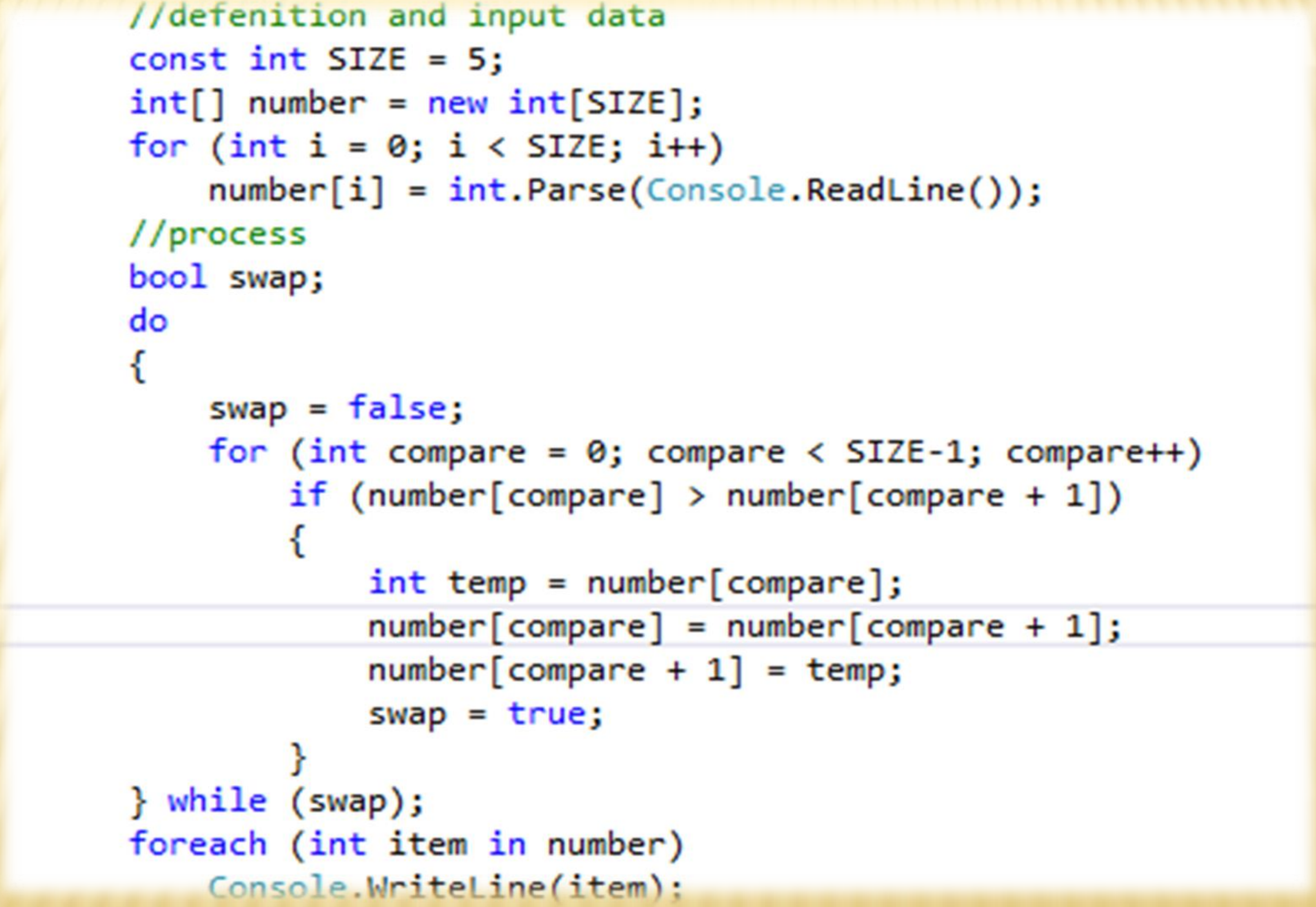
Pass	compares
SIZE-1	SIZE-1
SIZE-2	SIZE-2
SIZE-2	SIZE-3
1	1
0	0

بیشترین پنج عدد(BUBBLE)



```
//defenition and input data
const int SIZE = 5;
int[] number = new int[SIZE];
for (int i = 0; i < SIZE; i++)
    number[i] = int.Parse(Console.ReadLine());
//process
for (int pass = SIZE-1; pass >0; pass--)
    for (int compare = 0; compare < pass; compare++)
        if (number[compare] > number[compare + 1])
        {
            int temp = number[compare];
            number[compare] = number[compare + 1];
            number[compare + 1] = temp;
        }
foreach (int item in number)
    Console.WriteLine(item);
```

بیشترین پنج عدد(BUBBLE)



```
//defenition and input data
const int SIZE = 5;
int[] number = new int[SIZE];
for (int i = 0; i < SIZE; i++)
    number[i] = int.Parse(Console.ReadLine());
//process
bool swap;
do
{
    swap = false;
    for (int compare = 0; compare < SIZE-1; compare++)
        if (number[compare] > number[compare + 1])
        {
            int temp = number[compare];
            number[compare] = number[compare + 1];
            number[compare + 1] = temp;
            swap = true;
        }
} while (swap);
foreach (int item in number)
    Console.WriteLine(item);
```

ADA LOVELACE

?

ADA LOVELACE

- اوگوستا آدا بایرون (لاولیس)، دختر لرد بایرون، شاعر معروف انگلیسی، پدر برنامه نویسی است!
- شهرت آدا بیشتر به واسطه نوشتن توضیحاتی در مورد ماشین محاسباتی مکانیکی «چارلز بابیج» است. چارلز بابیج، یک فیلسوف تحلیلگر و ریاضیدان انگلیسی و نخستین کسی بود که ایده یک ماشین محاسبه گر برنامه‌پذیر را ارائه داد. وی در سال ۱۸۲۰ طراحی ماشین محاسبه گر خود را آغاز نمود ؛ ماشینی که بعدها به نام ماشین تفاضلی معروف شد. بابیج در زمان حیاتش هیچگاه نتوانست نتیجه کار ماشین محاسبه گر خود را ببیند، اما توانست امکانپذیر بودن برخی گزینه های بنظر نا ممکن را اثبات کند.

با تشکر از توجه شما

CsharpSchool.blog.ir

Flowchart.blog.ir

AbuzarArtSchool@yahoo.com

پایان