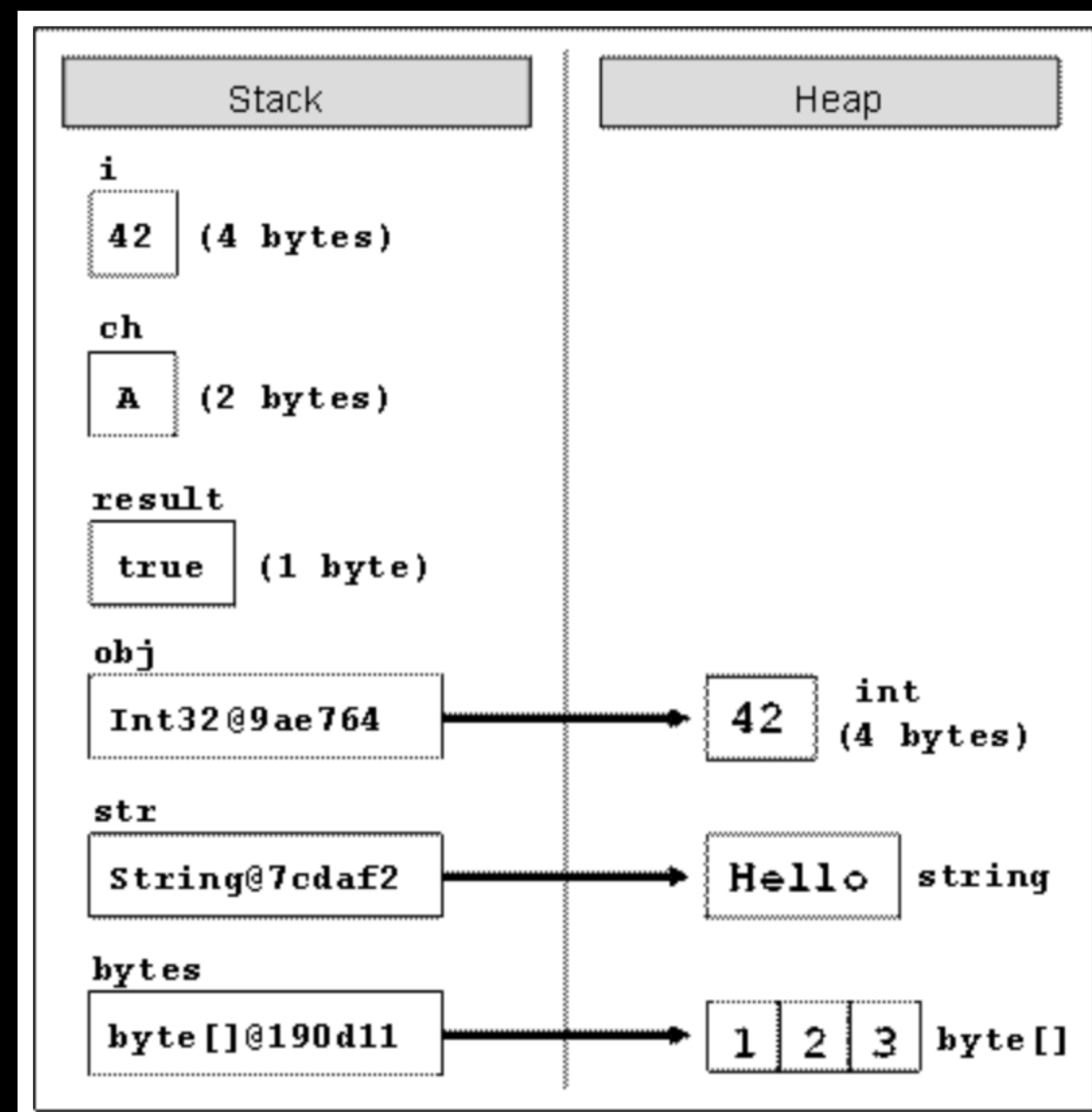


Паралелно програмиране

Упражнение 1 Въведение в C# и използване на
Parallel.For конструкция

Типове данни

Тип данни	Стойност по подразбиране	Минимална стойност	Максимална стойност
sbyte	0	-128	127
byte	0	0	255
short	0	-32768	32767
ushort	0	0	65535
int	0	-2147483648	2147483647
uint	0u	0	4294967295
long	0L	-9223372036854775808	9223372036854775807
ulong	0u	0	18446744073709551615
float	0.0f	$\pm 1.5 \times 10^{-45}$	$\pm 3.4 \times 10^{38}$
double	0.0d	$\pm 5.0 \times 10^{-324}$	$\pm 1.7 \times 10^{308}$
decimal	0.0m	$\pm 1.0 \times 10^{-28}$	$\pm 7.9 \times 10^{28}$
boolean	false	Възможните стойности са две – true или false	
char	'\u0000'	'\u0000'	'\uffff'
object	null	-	-
string	null	-	-



Оператори

Приоритет	Оператори
най-висок	++, -- (като постфикс), new, (type), typeof, sizeof
	++, -- (като префикс), +, - (едноаргументни), !, ~
	*, /, %
	+ (свързване на низове)
	+, -
	<<, >>
...	<, >, <=, >=, is, as
	==, !=
	&, ^,
	&&
	?:, ??
най-нисък	=, *=, /=, %=, +=, -=, <<=, >>=, &=, ^=, =

УСЛОВНИ КОНСТРУКЦИИ

```
if (булев израз)
{
    тяло на условната конструкция;
}
```

```
if (булев израз)
{
    тяло на условната конструкция;
}
else
{
    тяло на else-конструкция;
}
```

```
switch (селектор)
{
    case целочислена-стойност-1: конструкция; break;
    case целочислена-стойност-2: конструкция; break;
    case целочислена-стойност-3: конструкция; break;
    case целочислена-стойност-4: конструкция; break;
    // ...
    default: конструкция; break;
}
```

Масиви

```
int[] myArray = { 1, 2, 3, 4, 5, 6 };
```

```
string[] daysOfWeek =  
    { "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday",  
      "Sunday" };
```

```
int[] myArray = new int[6];  
myArray[1] = 1;  
myArray[5] = 5;
```

Матрици

```
Console.Write("Enter the number of the rows: ");
int rows = int.Parse(Console.ReadLine());

Console.Write("Enter the number of the columns: ");
int cols = int.Parse(Console.ReadLine());

int[,] matrix = new int[rows, cols];

Console.WriteLine("Enter the cells of the matrix:");

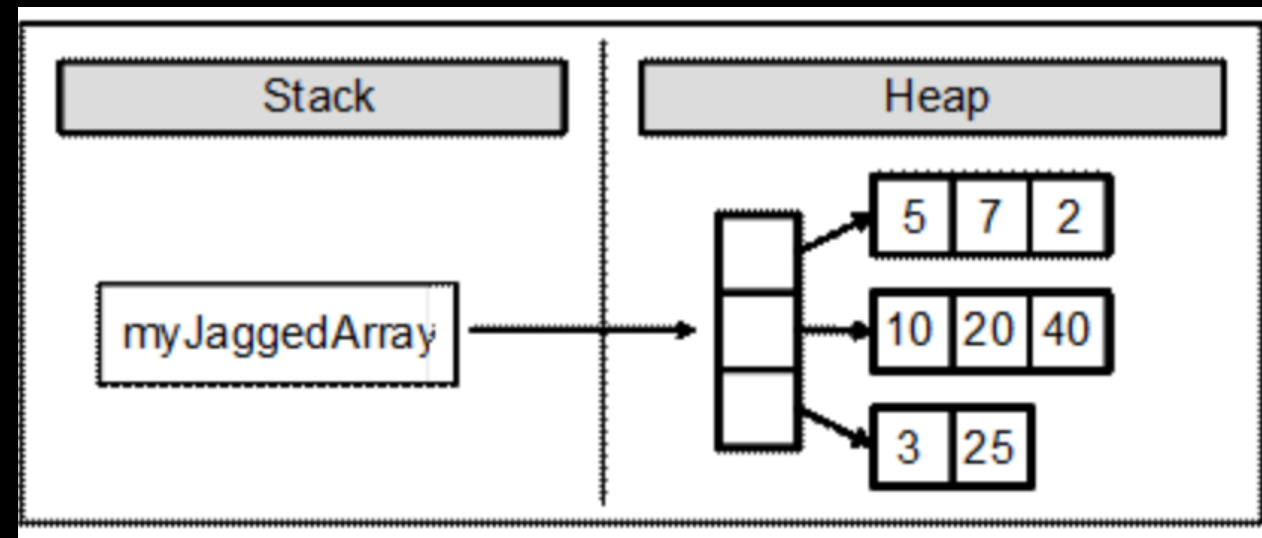
for (int row = 0; row < rows; row++)
{
    for (int col = 0; col < cols; col++)
    {
        Console.Write("matrix[{0},{1}] = ", row, col);
        matrix[row, col] = int.Parse(Console.ReadLine());
    }
}

for (int row = 0; row < matrix.GetLength(0); row++)
{
    for (int col = 0; col < matrix.GetLength(1); col++)
    {
        Console.Write(" " + matrix[row, col]);
    }
    Console.WriteLine();
}
```

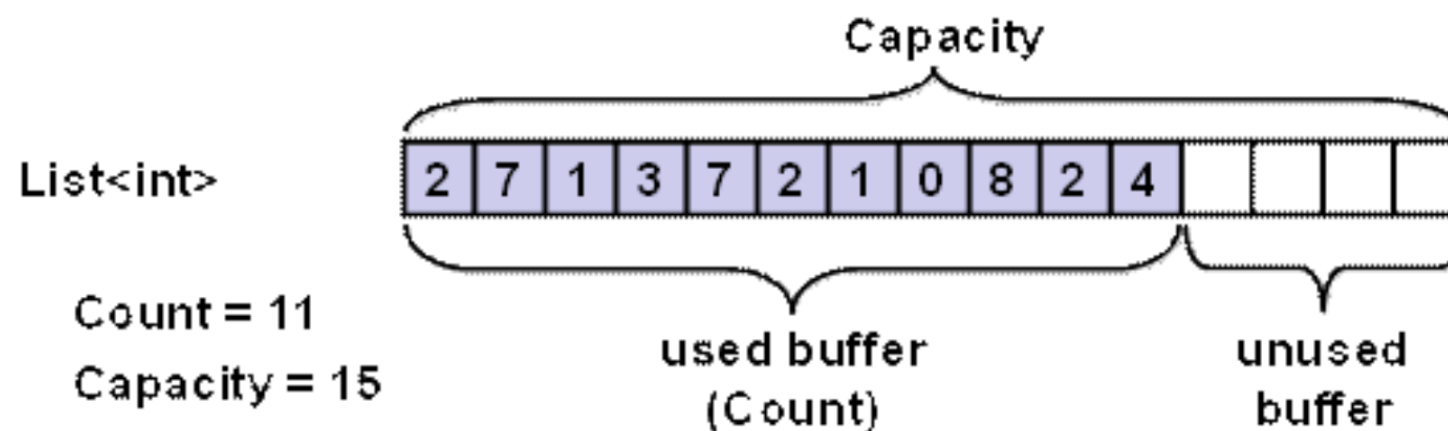
```
int[][] jaggedArray;
jaggedArray = new int[2][];
jaggedArray[0] = new int[5];
jaggedArray[1] = new int[3];
```

```
int[][] myJaggedArray = {
    new int[] {5, 7, 2},
    new int[] {10, 20, 40},
    new int[] {3, 25}
};
```

```
int[][][] jaggedOfMulti = new int[2][,];
jaggedOfMulti[0] = new int[,] { { 5, 15 }, { 125, 206 } };
jaggedOfMulti[1] = new int[,] { { 3, 4, 5 }, { 7, 8, 9 } };
```



СПИСЪК



```
using System;
using System.Collections;

class ProgrArrayListExample
{
    public static void Main()
    {
        ArrayList list = new ArrayList();
        list.Add("Hello");
        list.Add(5);
        list.Add(3.14159);
        list.Add(DateTime.Now);

        for (int i = 0; i < list.Count; i++)
        {
            object value = list[i];
            Console.WriteLine(
                "Index={0}; Value={1}\n", i, value);
        }
    }
}
```

Цикълът For

```
for (инициализация; условие; обновяване)
{
    тяло на цикъла;
}
```

```
int[] arr = new int[5];
for (int i = 0; i < arr.Length; i++)
{
    arr[i] = i;
}
```

```
for (int i = 0; i <= 10; i++)
{
    Console.Write(i + " ");
}
```


Цикъл foreach

```
foreach (variable in collection)
{
    statements;
}
```

```
int[] numbers = { 2, 3, 5, 7, 11, 13, 17, 19 };
foreach (int i in numbers)
{
    Console.Write(" " + i);
}
Console.WriteLine();
String[] towns = { "Sofia", "Plovdiv", "Varna", "Bourgas" };
foreach (String town in towns)
{
    Console.Write(" " + town);
}
```

```

using System;
using System.Diagnostics;
using System.Threading.Tasks;

namespace ParallelPrograming
{
    class Program
    {
        public static void Main(string[] args)
        {
            var cores = Environment.ProcessorCount;

            Stopwatch sw = new Stopwatch();

            // create the arrays to hold the data and the results
            int[] dataItems = new int[10000000];
            double[] resultItems = new double[10000000];

            // create the data items
            for (int i = 0; i < dataItems.Length; i++) {
                dataItems[i] = i;
            }

            sw.Start();
            // process the data in a parallel for loop
            for (int index = 0; index < dataItems.Length; index++) {
                resultItems[index] = Math.Pow(dataItems[index], 2);
            }
            sw.Stop();
            Console.WriteLine(sw.ElapsedMilliseconds);
            sw.Reset();

            sw.Start();
            // process the data in a parallel for loop
            Parallel.For(0, dataItems.Length, (index) => {
                resultItems[index] = Math.Pow(dataItems[index], 2);
            });
            sw.Stop();
            Console.WriteLine(sw.ElapsedMilliseconds);

            // wait for input before exiting
            Console.WriteLine("Press enter to finish");
            Console.ReadLine();
        }
    }
}

```

For и Parallel.For

```

static void Main()
{
    for (int i = 0; i < length; i++)
    {
    }
}

```

```

static void Main()
{
    Parallel.For(from, to, i =>
    {
    }));
}

```

Задачи

Задача 1 Да се тестват всички задачи от презентацията в желана от вас среда.

Задача 2 Да се намерят всички прости числа в интервала от 1 до 1000000. Да се реализира задачата като се използва последователен For цикъл и след това да се повтори и с Parallel.For.

Задача 3 Да се намери произведението на две матрици със следният брой елементи 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800 и 2000. Да се реализира задачата като се използва последователен For цикъл и след това да се повтори и с Parallel.For. Принтирайте и времето за което са приключили и двете задачи.