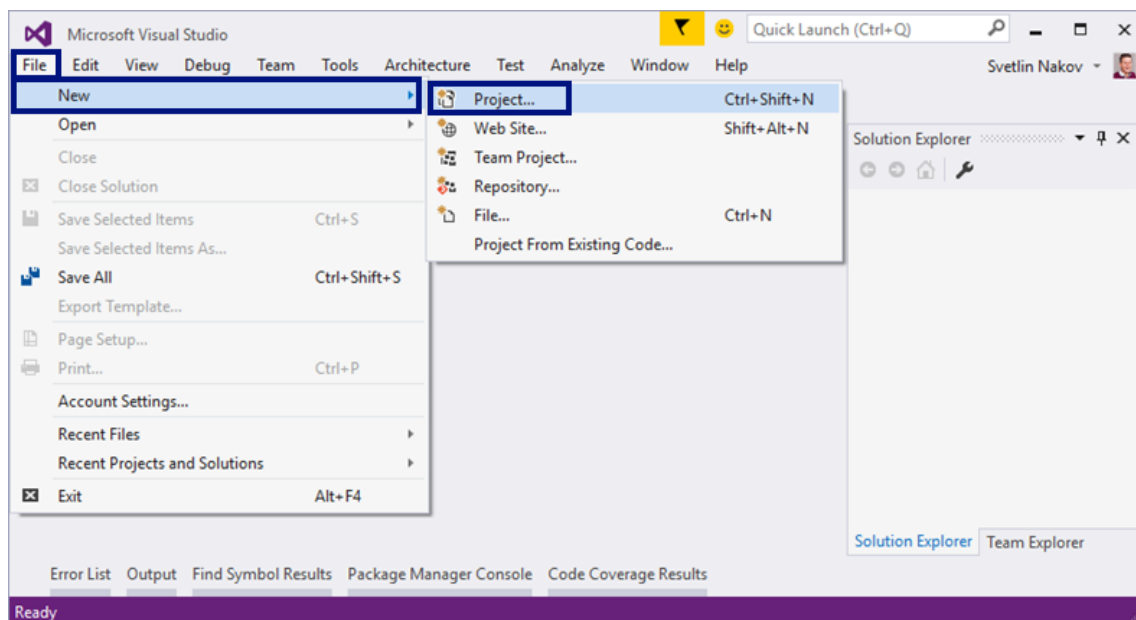


# Упражнения: Списъци – обхождания

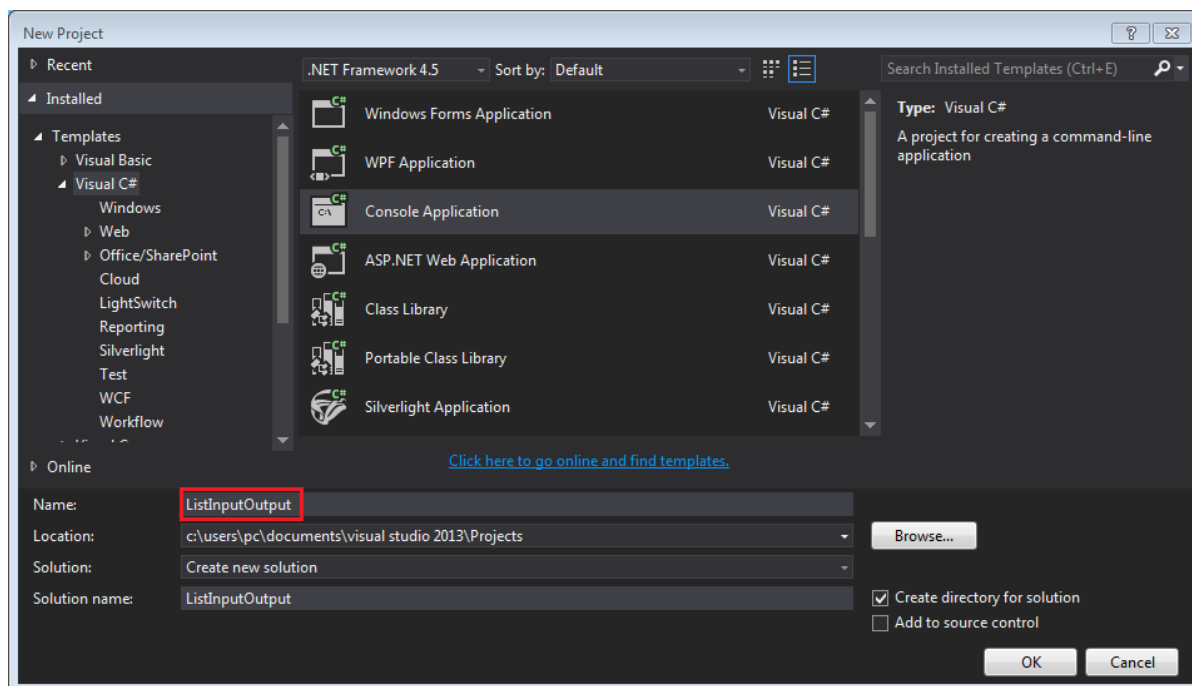
## 1. Въвеждане на списък от конзолата

Въведете **списък от цели числа** и го изведете в конзолата

1. Стартирайте Visual Studio.
2. Създайте нов конзолен проект: [File] → [New] → [Project].



3. Изберете от диалоговия прозорец [Visual C#] → [Windows] → [Console Application] и дайте подходящо име на проекта, например **“ListInputOutput”**:

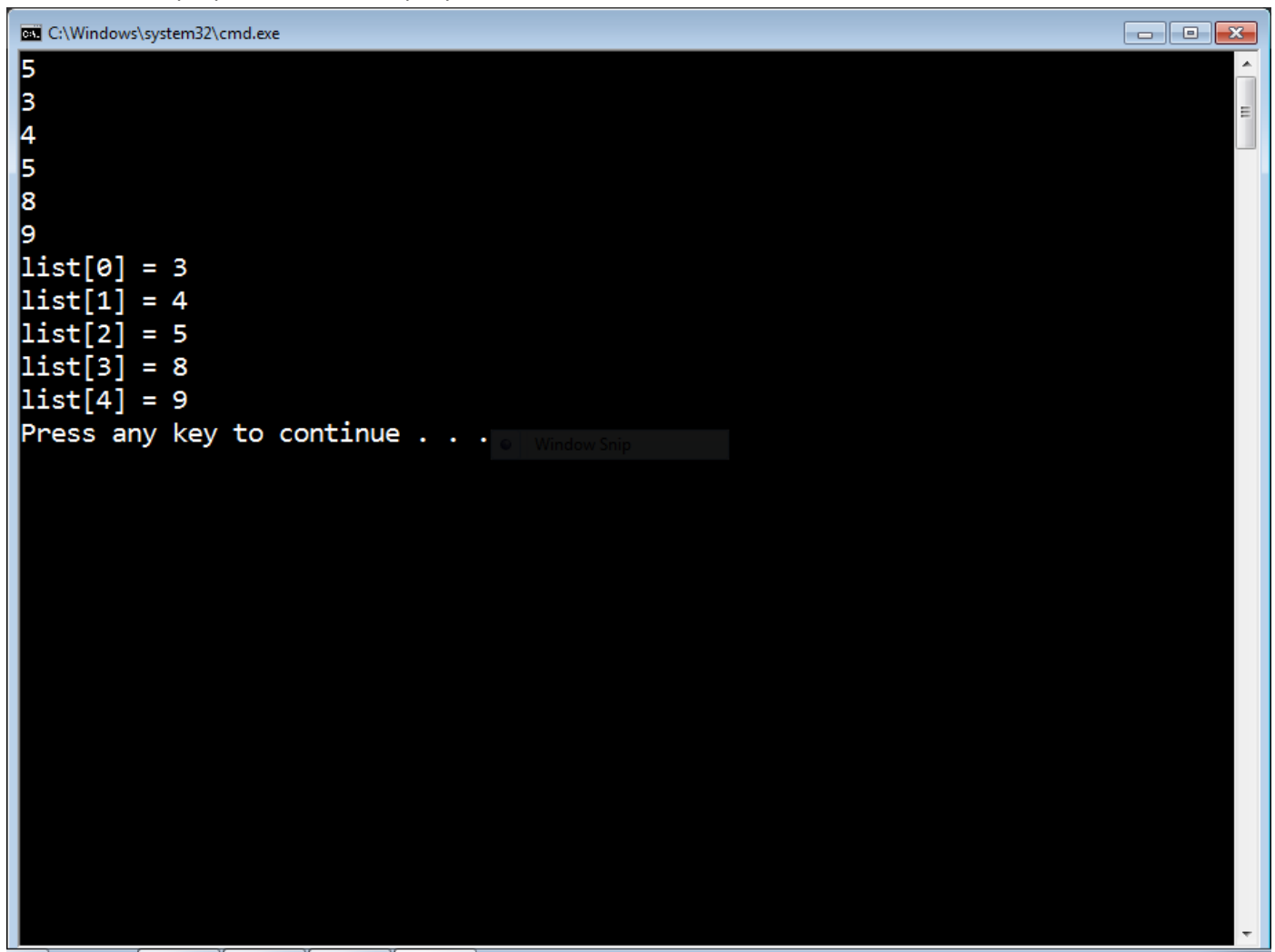


4. Намерете секцията `Main(string[] args)`. В нея напишете следния програмен код:

```
int n = int.Parse(Console.ReadLine());
List<int> list = new List<int>();
for (int i = 0; i < n; i++) {
    list.Add(int.Parse(Console.ReadLine()));
}

for (int index = 0; index < list.Count; index++) {
    Console.WriteLine("list[{0}] = {1}", index, list[index]);
}
```

5. **Стартирайте** програмата с натискане на **[Ctrl+F5]**.
6. Въведете **брой на елементите** в списъка – цяло число.
7. Въведете на **отделни редове** елементите за списъка.
8. Очаквайте от програмата подобен резултат:



```
C:\Windows\system32\cmd.exe
5
3
4
5
8
9
list[0] = 3
list[1] = 4
list[2] = 5
list[3] = 8
list[4] = 9
Press any key to continue . . .
```

## 2. Списък от четни числа

Въведете **списък от цели числа** и изведете **четните** числа от списъка на един ред в конзолата. Елементите на списъка ще получите от единствен ред, разделени с интервали.

### Примери

Вход	Изход
3 4 8 5 7 5 2 1	4 8 2
1 2 4 3	2 4
7 2 8 3 5 9 7 3	2 8

## 3. Списък от крайности

Въведете **списък от цели числа** и изведете тези от тях, които са равни на **минималния или максималния** елемент.

### Примери

Вход	Изход
5 4 8 5 7 8 2 1	1 8 8
1 1 1	1 1 1
4 2 8 3 5 9 2 3	2 2 9

## 4. Максимална поредица еднакви числа

Въведете **списък от цели числа** и намерете **най-дългата поредица от еднакви елементи**. Ако съществуват няколко, изпечатайте **най-лявата**.

### Примери

Вход	Изход
3 4 4 5 5 5 2 2	5 5 5
7 7 4 4 5 5 3 3	7 7
1 2 3 3	3 3

### Подказки

- Обходете позициите **p** отляво надясно и пазете **началото** и **дължината** на текущата поредица от еднакви числа приключваща с **p**.
- Също така пазете текущата най-добра (най-дълга) поредица (**bestStart** – позицията, на която започва, както и **bestLength** – нейната дължина) и я обновявайте след всяка стъпка

## 5. Сума на обърнати числа

Напишете програма, която прочита поредица от цели числа, преобръща техните цифри и ги сумира.

## Примери

Вход	Изход	Пояснения
123 234 12	774	$321 + 432 + 21 = 774$
12 12 34 84 66 12	220	$21 + 21 + 43 + 48 + 66 + 21 = 220$
120 1200 12000	63	$21 + 21 + 21 = 63$