

# Лаб: Вложени цикли

Задачи за упражнение в клас и за домашно към курса "[Основи на програмирането](#)" @ СофтУни.

## 1. Числата от N до 1 в обратен ред

Напишете програма, която чете цяло положително число **n**, въведено от потребителя, и печата **числата от n до 1 в обратен ред** (от най-голямото към най-малкото).

### Примерен вход и изход

вход	изход	вход	изход	вход	изход
2	2 1	3	3 2 1	5	5 4 3 2 1

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Compete/Index/1016#0>

Насоки:

1. Прочетете **едно цяло число** от конзолата – **n**:

```
int n = int.Parse(Console.ReadLine());
```

2. Направете **един for цикъл** от **n** до **0**, но вместо да увеличавате променливата **i** с **1**, я намаляйте с **1** при всяка итерация на цикъла:

```
for (int i = n; i > 0; i--)  
{  
    ...  
}
```

3. В тялото на цикъла **принтирайте** променливата **i**:

```
for (int i = n; i > 0; i--)  
{  
    Console.WriteLine(i);  
}
```

## 2. Числата от 1 до N през 3

Напишете програма, която чете число **n**, въведено от потребителя, и отпечатва **числата от 1 до n през 3** (със стъпка 3).

## Примерен вход и изход

вход	изход
10	1 4 7 10

вход	изход
7	1 4 7

вход	изход
15	1 4 7 10 13

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Compete/Index/1016#1>

Насоки:

1. Прочетете входните данни от конзолата – едно цяло число  $n$

```
int n = int.Parse(Console.ReadLine());
```

2. Направете for цикъл от 1 до  $n$  (включително) и задайте стъпка 3. Това означава, че при всяка итерация на цикъла, променливата  $i$  ще увеличава стойността си с 3, вместо с 1:

```
for (int i = 1; i <= n; i += 3)
{
    ...
}
```

3. В тялото на цикъла принтирайте променливата  $i$ :

```
for (int i = 1; i <= n; i += 3)
{
    Console.WriteLine(i);
}
```

## 3. Четни степени на 2

Да се напише програма, която чете число  $n$ , въведено от потребителя, и печата четните степени на  $2 \leq 2^n$ :  $2^0$ ,  $2^2$ ,  $2^4$ ,  $2^8$ , ...,  $2^n$ .

## Примерен вход и изход

вход	изход
3	1 4

вход	изход
4	1 4 16

вход	изход
5	1 4 16

вход	изход
6	1 4 16 64

вход	изход
7	1 4 16 64

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Compete/Index/1016#2>

Насоки:



1. Прочетете **входните данни** от конзолата – **едно цяло число n**:

```
int n = int.Parse(Console.ReadLine());
```

2. **Инициализирайте** помощна **променлива** и й **задайте първоначална стойност 1** (Ще я използваме за да печатаме една степен на числото 2 при всяка итерация на цикъла).

```
int current = 1;
```

3. Направете **for** **цикъл** от **0** до **n** (**включително**) и за да отпечатате **само четните степени на 2**, му задайте **стъпка 2**:

```
for (int i = 0; i <= n; i += 2)
{
    ...
}
```

4. В **тялото на цикъла**, **принтирайте** помощната **променлива**, която **създадохме**, след което я **умножавайте по 4**:

```
for (int i = 0; i <= n; i += 2)
{
    Console.WriteLine(current);
    current = current * 4;
}
```

## 4. Сграда

Напишете програма, която извежда на конзолата номерата на стаите в една сграда (в низходящ ред), като са изпълнени следните условия:

- На **всеки четен етаж** има **само офиси**
- На **всеки нечетен етаж** има **само апартаменти**
- Всеки **апартамент** се означава по следния начин : **A{номер на етаж}{номер на апартамента}**, номерата на апартаментите започват от **0**.
- Всеки **офис** се означава по следния начин : **O{номер на етаж}{номер на офиса}**, номерата на офисите също започват от **0**.
- На **последният етаж** винаги има **апартаменти** и те са по-големи от останалите, за това **пред номера** им пише **'L'**, вместо **'A'**. Ако има само един етаж, то има само големи апартаменти!

От конзолата се прочитат две **цели числа** - **броят на етажите** и **броят на стаите за един етаж**.

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Compete/Index/1016#3>

## Примерен вход и изход

Вход	Изход	Обяснения	
6 4	L60 L61 L62 L63 A50 A51 A52 A53 O40 O41 O42 O43 A30 A31 A32 A33 O20 O21 O22 O23 A10 A11 A12 A13	Имаме общо 6 етажа, с по 4 стаи на етаж. Нечетните етажи имат само апартаменти, а четните само офиси.	
Вход	Изход	Вход	Изход
9 5	L90 L91 L92 L93 L94 O80 O81 O82 O83 O84 A70 A71 A72 A73 A74 O60 O61 O62 O63 O64 A50 A51 A52 A53 A54 O40 O41 O42 O43 O44 A30 A31 A32 A33 A34 O20 O21 O22 O23 O24 A10 A11 A12 A13 A14	4 4	L40 L41 L42 L43 A30 A31 A32 A33 O20 O21 O22 O23 A10 A11 A12 A13

## 5. Комбинации

Напишете програма, която изчислява **колко решения в естествените числа** (включително и нулата) има уравнението:

$$x_1 + x_2 + x_3 + x_4 + x_5 = n$$

Числото  $n$  е цяло число и се въвежда от конзолата.

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Compete/Index/1016#4>

## Примерен вход и изход

Вход	Изход	Обяснения	Вход	Изход	Вход	Изход
25	23751	Генерираме всички комбинации от 5 числа, като първата е: $0+0+0+0+0=0$ , но понеже не е равна на 25, продължаваме: $0+0+0+0+1=1$ – също не е 25 и т.н Стигаме до първата валидна комбинация: $0+0+0+0+25=25$ , увеличаваме броя на валидни комбинации с 1, втората валидна комбинация е: $0+0+0+1+24=25$ Третата: $0+0+0+2+23=25$ и т.н След генериране на всички възможни комбинации, броят на валидните е 23751.	20	10626	5	126

## 6. Пътуване

Ани обича да пътува и иска тази година да посети няколко различни дестинации. Като си избере дестинация, ще прецени колко пари ще ѝ трябват за да отиде до там и ще започне да спестява. Когато е спестила достатъчно, ще може да пътува.

От конзолата всеки път ще се четат първо дестинацията и минималния бюджет, който ще е нужен за пътуването .

След това ще се четат няколко суми, които Ани спестява като работи и когато успее да събере достатъчно за пътуването, ще заминава, като на конзолата трябва да се изпише:

"Going to {дестинацията}!"

Когато е посетила всички дестинации, които иска, вместо дестинация ще въведе "End" и програмата ще приключи.

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Compete/Index/1016#5>

## Примерен вход и изход

Вход	Изход	Вход	Изход
Greece 1000 200 200 300 100 150 240 Spain 1200 300 500 193 423 End	Going to Greece! Going to Spain!	France 2000 300 300 200 400 190 258 360 Portugal 1450 400 400 200 300 300 Egypt 1900 1000 280 300 500 End	Going to France! Going to Portugal! Going to Egypt!

## 7. Война на имена

Напишете програма, която изчислява ASCII стойността на няколко имена, като името с най-голяма стойност е победител. Стойността на името се изчислява като съберем ASCII стойностите на всички букви. От конзолата ще се четат имена до получаването на команда „STOP“, след което трябва да се изпише:

"Winner is {името на победителя} – {стойността на името му}!"



## Примерен вход и изход

Вход	Изход	Обяснения
Petar Georgi Stanimir STOP	Winner is Stanimir - 839!	Първата буква е <b>P</b> и тя отговаря на 80 в ASCII, <b>e</b> отговаря на 101, <b>t</b> отговаря на 116, <b>a</b> отговаря на 97, <b>r</b> отговаря на 114. Сумата им е 508. Продължаваме да правим същите изчисления и за останалите и получаваме, че името на Stanimir има най-голяма стойност – 839.
Ivo Niki Valio Konstantin STOP	Winner is Konstantin - 1065!	

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Compete/Index/1016#6>

## 8. Фабрика за бисквити

Фабрика за бисквити приема поръчки всеки ден. Напишете програма, която помага на сладкарите да направят бисквитите по-бързо, само като въвеждат необходимите продукти в компютъра.

Основните компоненти са **брашно, яйца и захар** и те **винаги трябва да присъстват** в сместа. За различните видове бисквити се прибавят различни допълнителни продукти, **техния брой не е ограничен**. Като **вход програмата първо ще приема едно цяло число - броят на партидите, които трябва да се направят днес**. На следващите редове ще се въвеждат продуктите за всяка смес. **При въвеждане на команда "Bake!",** съответната смес ще се слага във фурната.

Ако сместа за печене **не съдържа поне един от задължителните компоненти : брашно, яйца или захар**, то трябва да се изписва : "The batter should contain flour, eggs and sugar!", а **ако съдържа всички компоненти** : "Baking batch number {номера на партидата, която печем} ...".

Тествайте решението си в judge системата: <https://judge.softuni.bg/Contests/Compete/Index/1016#7>

## Примерен вход и изход

Вход	Изход	Вход	Изход
------	-------	------	-------

2 flour eggs sugar chocolate Bake! flour eggs sugar caramel peanuts Bake!	Baking batch number 1... Baking batch number 2...	3 flour eggs jam Bake! sugar Bake! flour eggs milk almonds sugar Bake! flour eggs sugar Bake!	The batter should contain flour, eggs and sugar! Baking batch number 1... Baking batch number 2... Baking batch number 3...
--	--	---	--

## Примерни изпитни задачи

### Магически числа

Тествайте решението си [тук](#).

Да се напише програма, която чете едно цяло „магическо“ число, въведено от потребителя, и изкарва **всички** възможни **6-цифрени числа**, за които **произведението на неговите цифри е равно** на „магическото“ число.

Пример: „Магическо число“ -> 2

- 111112 ->  $1 * 1 * 1 * 1 * 1 * 2 = 2$
- 111121 ->  $1 * 1 * 1 * 1 * 2 * 1 = 2$
- 111211 ->  $1 * 1 * 1 * 2 * 1 * 1 = 2$
- 112111 ->  $1 * 1 * 2 * 1 * 1 * 1 = 2$
- 121111 ->  $1 * 2 * 1 * 1 * 1 * 1 = 2$
- 211111 ->  $2 * 1 * 1 * 1 * 1 * 1 = 2$

### Вход

Входът се чете от конзолата и се състои от **едно цяло число** в интервала [1...600000].

### Изход

На конзолата трябва да се отпечата **всички** „магически“ числа, разделени с **интервал**.

### Примерен вход и изход

вход	изход
2	111112 111121 111211 112111 121111 211111
8	111118 111124 111142 111181 111214 111222 111241 111412 111421 111811 112114 112122 112141 112212 112221 112411 114112 114121 114211 118111 121114 121122 121141 121212 121221 121411 122112 122121 122211 124111 141112 141121 141211 142111 181111 211114 211122 211141 211212 211221 211411 212112 212121 212211 214111 221112 221121 221211 222111 241111 411112 411121 411211 412111 421111 811111

531441	999999
--------	--------