

# 1. Лаб: Вложени цикли

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

Тествайте решенията си в Judge системата <https://judge.softuni.bg/Contests/2421>

## 1. Часовник

Напишете програма, която отпечатва **часовете в денонощието от 0:0 до 23:59**, всеки на отделен ред. Часовете трябва да се изписват във формат "{час} : {минути}" .

### Примерен вход и изход

Вход	Изход
(няма вход)	0:0 0:1 0:2 0:3 0:4 0:5 0:6 0:7 0:8 0:9 0:10 ... 23:50 23:51 23:52 23:53 23:54 23:55 23:56 23:57 23:58 23:59

### Насоки

1. Създайте 2 вложени for-цикъла, с които да итерирате през всяка една минута и час от денонощието:

```
for h in range(24):  
    for m in range(60):
```

2. Отпечатайте резултата:

```
for h in range(24):  
    for m in range(60):  
        print(f"{h}:{m}")
```

## 2. Таблица за умножение

Отпечатайте на конзолата таблицата за умножение за числата от 1 до 10 във формат:

"{първи множител} \* {втори множител} = {резултат}".

### Примерен вход и изход

Вход	Изход
(няма вход)	1 * 1 = 1 1 * 2 = 2 1 * 3 = 3 1 * 4 = 4 1 * 5 = 5 1 * 6 = 6 1 * 7 = 7 1 * 8 = 8 1 * 9 = 9 1 * 10 = 10 ... 10 * 1 = 10 10 * 2 = 20 10 * 3 = 30 10 * 4 = 40 10 * 5 = 50 10 * 6 = 60 10 * 7 = 70 10 * 8 = 80 10 * 9 = 90 10 * 10 = 100

### Насоки

1. Създайте 2 вложени for-цикъла, с които да итерирате всяка възможна стойност на двата множителя от 1 до 10:

```
for x in range(1, 11):  
    for y in range(1, 11):
```

2. Намерете произведението на двата множителя и отпечатайте резултата:

```

for x in range(1, 11):
    for y in range(1, 11):
        product = x * y
        print(f"{x} * {y} = {product}")

```

### 3. Комбинации

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

$$x_1 + x_2 + x_3 = n$$

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

### Примерен вход и изход

Вход	Изход	Обяснения	Вход	Изход	Вход	Изход
25	351	Генерираме всички комбинации от 5 числа, като първата е: $0+0+0=0$ , но понеже не е равна на 25, продължаваме: $0+0+1=1$ – също не е 25 и т.н Стигаме до първата валидна комбинация: $0 + 0 + 25 = 25$ , увеличаваме броя на валидни комбинации с 1, втората валидна комбинация е: $0 + 1 + 24 = 25$ Третата: $0 + 2 + 23 = 25$ и т.н След генериране на всички възможни комбинации, броят на валидните е 351.	20	231	5	21

### Насоки

1. Прочетете входните данни – **едно цяло число  $n$** , въведено от потребителя, и го запаметете в променлива;
2. Създайте 3 вложени for цикъла, с които да итерирате всяка възможна стойност на едно от 3-те числа в уравнението:

```

n = int(input())
for x1 in range(0, n + 1):
    for x2 in range(0, n + 1):
        for x3 in range(0, n + 1):

```

3. Направете проверка в най-вътрешния вложен цикъл за стойностите на  **$x_1$ ,  $x_2$ ,  $x_3$**  във всяка една итерация. За да бъде валидно уравнението, техният сбор трябва да е равен на  **$n$** . Създайте променлива **counter**, която да пази броя на валидните комбинации и добавяйте към нея всеки път, когато генерирате такава:

```

n = int(input())
combinations_count = 0
for x1 in range(0, n + 1):
    for x2 in range(0, n + 1):
        for x3 in range(0, n + 1):
            if x1 + x2 + x3 == n:
                combinations_count += 1

```

4. Накрая принтирайте броя на валидните комбинации.

## 4. Сума от две числа

Напишете програма която проверява **всички възможни комбинации** от двойка числа в интервала от две дадени числа. На изхода се отпечатва, **коя** **пored** е комбинацията чиито **сбор** от числата е **равен** на дадено **магическо число**. Ако няма **нито една комбинация** отговаряща на условието се отпечатва **съобщение**, че не е намерено.

### Вход

Входът се чете от конзолата и се състои от **три реда**:

- **Първи ред** – начало на интервала – цяло число в интервала [1...999]
- **Втори ред** – край на интервала – цяло число в интервала [по-голямо от първото число...1000]
- **Трети ред** – магическото число – цяло число в интервала [1...10000]

### Изход

На конзолата трябва да се отпечата **един ред**, според резултата:

- Ако е **намерена комбинация** чиито **сбор** на числата е **равен** на магическото число
  - "Combination N:{пореден номер} ({първото число} + {второ число} = {магическото число})"
- Ако **не е намерена комбинация** отговаряща на условието
  - "{броят на всички комбинации} combinations - neither equals {магическото число}"

### Примерен вход и изход

Вход	Изход	Обяснения	Вход	Изход
1 10 5	Combination N:4 (1 + 4 = 5)	Всички комбинации от две числа между 1 и 10 са: 1 1, 1 2, 1 3, <b>1 4</b> , 1 5, ... 2 1, 2 2, ... 4 9, 4 10, 5 1 ... 10 9, 10 10 Първата комбинация, чиито сбор на числата е равен на магическото число 5 е <b>четвъртата (1 и 4)</b>	88 888 1000	Combination N:20025 (112 + 888 = 1000)
Вход	Изход	Обяснения	Вход	Изход
23 24 20	4 combinations - neither equals 20	Всички комбинации от две числа между 23 и 24 са: 23 23, 23 24, 24 23, 24 24 (общо 4) Няма двойки числа, чиито сбор е равен на магическото 20	88 888 2000	641601 combinations - neither equals 2000

## 5. Пътуване

Ани обича да пътува и иска тази година да посети **няколко** различни дестинации. Като си избере дестинация, ще прецени **колко пари ще й 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 (десетично число), който ще е нужен за пътуването.

След това ще се четат **няколко суми (десетични числа)**, 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.

### Примерен вход и изход

Вход	Изход	Вход	Изход
Greece	Going to Greece!	France	Going to France!
1000.00	Going to Spain!	2000.00	Going to Portugal!
200.00		300.00	Going to Egypt!
200.00		300.00	
300.00		200.00	
100.00		400.00	
150.00		190.00	
240.00		258.00	
Spain		360.00	
1200.00		Portugal	
300.00		1450.00	
500.00		400.00	
193.00		400.00	
423.00		200.00	
End		300.00	
		300.00	
		Egypt	
		1900.00	
		1000.00	
		280.00	
		300.00	
		500.00	
		End	

## 6. 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 : **О{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, затова **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.

## Примерен вход и изход

Вход	Изход	Обяснения	
<b>6</b> <b>4</b>	L60 L61 L62 L63 A50 A51 A52 A53 040 041 042 043 A30 A31 A32 A33 020 021 022 023 A10 A11 A12 A13	Имаме общо <b>6</b> етажа, с по <b>4</b> стаи на етаж. Нечетните етажи имат само апартаменти, а четните само офиси.	
Вход	Изход	Вход	Изход
9 5	L90 L91 L92 L93 L94 080 081 082 083 084 A70 A71 A72 A73 A74 060 061 062 063 064 A50 A51 A52 A53 A54 040 041 042 043 044 A30 A31 A32 A33 A34 020 021 022 023 024 A10 A11 A12 A13 A14	4 4	L40 L41 L42 L43 A30 A31 A32 A33 020 021 022 023 A10 A11 A12 A13