

Работа с вложени цикли

Цикли със стъпка, по-сложни задачи



СофтУни

Преподавателски екип



Software
University



SoftUni
Foundation



Софтуерен университет

<http://softuni.bg>

Имате въпроси?

sli.do

#pb-jan



Преговор

1. Колко пъти ще се изпише "SoftUni" на конзолата след изпълнението на следния код:

```
int i = 0;
while(i <= 5) {
    cout << "SoftUni" << endl;
    i++;
}
```

5

0

4

6

2. Колко пъти ще се изпише "SoftUni" на конзолата след изпълнението на следния код:

```
int i = 0;
while(i == 0) {
    cout << "SoftUni" << endl;
    if(i == 1)
        break;
}
```

0

1

100000

Безброй
много пъти

3. Колко итерации ще има следния цикъл:

```
int i = 0;  
while(i <= 10) {  
    i--;  
}
```

10

Безброй
много

11

2147483647

4. Какъв ще е резултатът от изпълнението на следния код:

```
int i = 0;
while (i < 6) {
    i++;
    if (i % 2 == 0)
        cout << i << endl;
}
```

024

24

246

123456



Вложени цикли

По-сложни комбинаторни задачи

Пример – часовник (1)

Часовете се променят
когато минутите
надвишат 59

19:03

Докато минутите се
променят часовете
остават същите



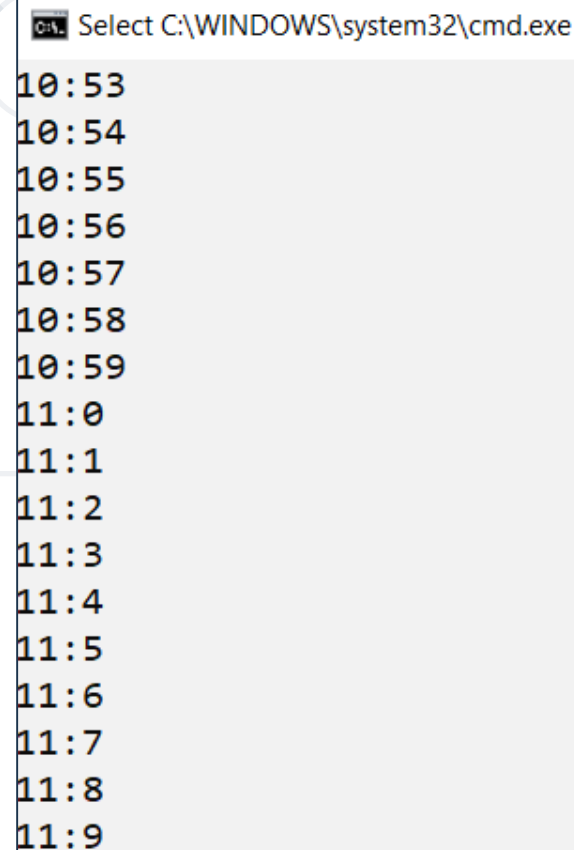
Как може да си направим часовник с код?

Демо

Пример – часовник (2)

- Външният цикъл отговаря за часовете, а вътрешния за минутите

```
for (int h = 0; h <= 23; h++) {  
    for (int m = 0; m <= 59; m++) {  
        cout << h << ":" << m << endl;  
    }  
}
```



```
Select C:\WINDOWS\system32\cmd.exe  
10:53  
10:54  
10:55  
10:56  
10:57  
10:58  
10:59  
11:0  
11:1  
11:2  
11:3  
11:4  
11:5  
11:6  
11:7  
11:8  
11:9
```

```
for (int i = 0; i < n; i++)  
    for (int j = 0; j < n; j++)  
        ...
```

Имената на
итераторите трябва
да бъдат различни

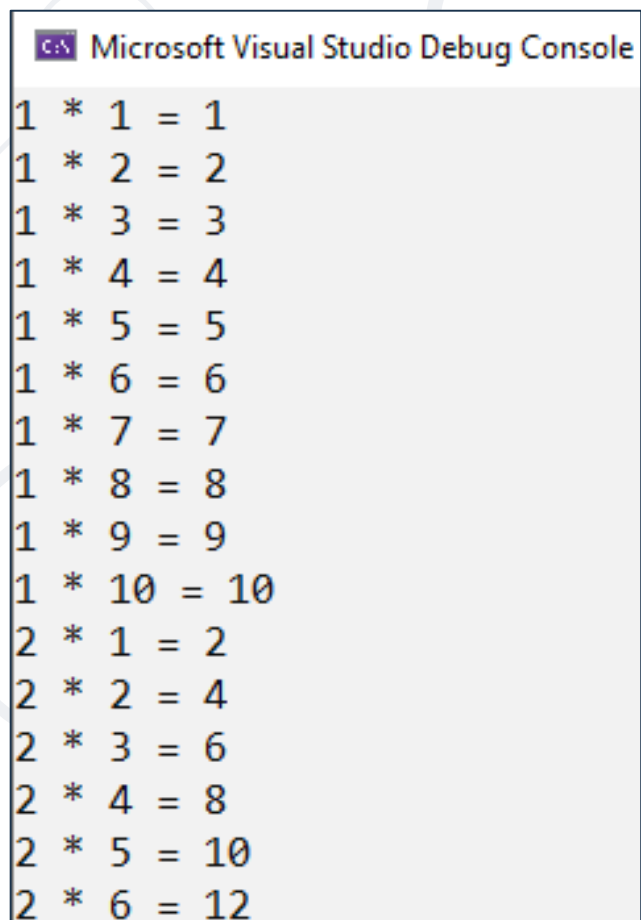
- За всяка итерация на външния цикъл вложения се изпълнява **n - на брой пъти**



Таблица за умножение - условие

- Отпечатайте на конзолата таблицата за умножение за числата от 1 до 10

- Изход:



```
Microsoft Visual Studio Debug Console
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
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
```

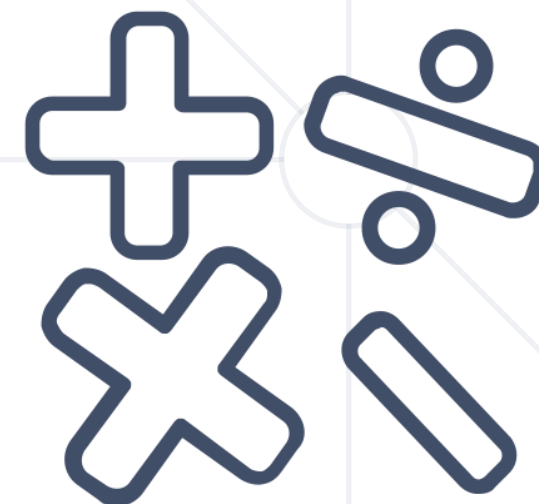


Таблица за умножение - решение

```
for (int x = 1; x <= 10; x++) {  
    for (int y = 1; y <= 10; y++) {  
        int product = x * y;  
        cout << x << " * " << y << " = " << product << endl;  
    }  
}
```

- За прекъсване на вложени цикли, използваме булеви променливи.

```
bool flag = false;
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++)
        if (condition)
            flag = true;
            break;
    if (flag)
        break;
```

Външният цикъл ще се прекъсне, само ако стойността на flag бъде true

- Напишете програма, която проверява всички възможни комбинации от двойка числа в даден интервал
 - Ако се намери комбинация, чийто **сбор от числата е равен** на дадено **магическо число** на изхода **се отпечатва съобщение** и програмата приключва изпълнение
 - Ако не се намери **ниито една комбинация**, отговаряща на условието се отпечатва **съобщение, че не е намерено**

Сума от две числа – условие (2)

- Примерен вход и изход:

1
10
5



Combination N:4 ($1 + 4 = 5$)

23
24
20



4 combinations - neither equals 20

Сума от две числа - решение

```
int startingNumber, finalNumber, magicNumber;
cin >> startingNumber >> finalNumber >> magicNumber;
int combinations = 0;
bool isFound = false;
for (int i = startingNumber; i <= finalNumber; i++) {
    for (int j = startingNumber; j <= finalNumber; j++) {
        combinations++;
        if (i + j == magicNumber) {
            cout << "Combination N:" << combinations
                << " (" << i << " + " << j << " = "
                << magicNumber << ")" << endl;
            isFound = true;
            break;
        }
    }
}
if (isFound)
    break;
} // Finish logic
```

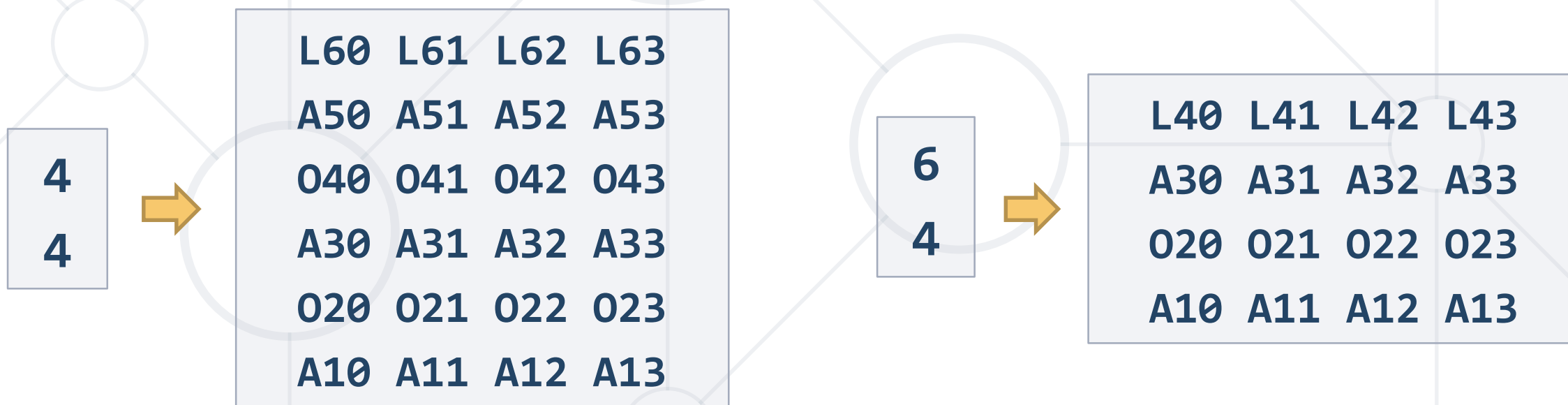
Ако намерим
комбинация, прекъсваме
вътрешният цикъл

- Напишете програма, която извежда номерата на стаите в една сграда (в низходящ ред)
 - На всеки **четен** етаж има само **офиси**
 - На всеки **нечетен** етаж има само **апартаменти**
- Етажите се означават по следния начин:
 - Апартаменти: "А{номер на етажа}{номер на апартамента}"
 - Офиси: "О{номер на етажа}{номер на офиса}"
 - Номерата им винаги започват с 0



Сграда – условие (2)

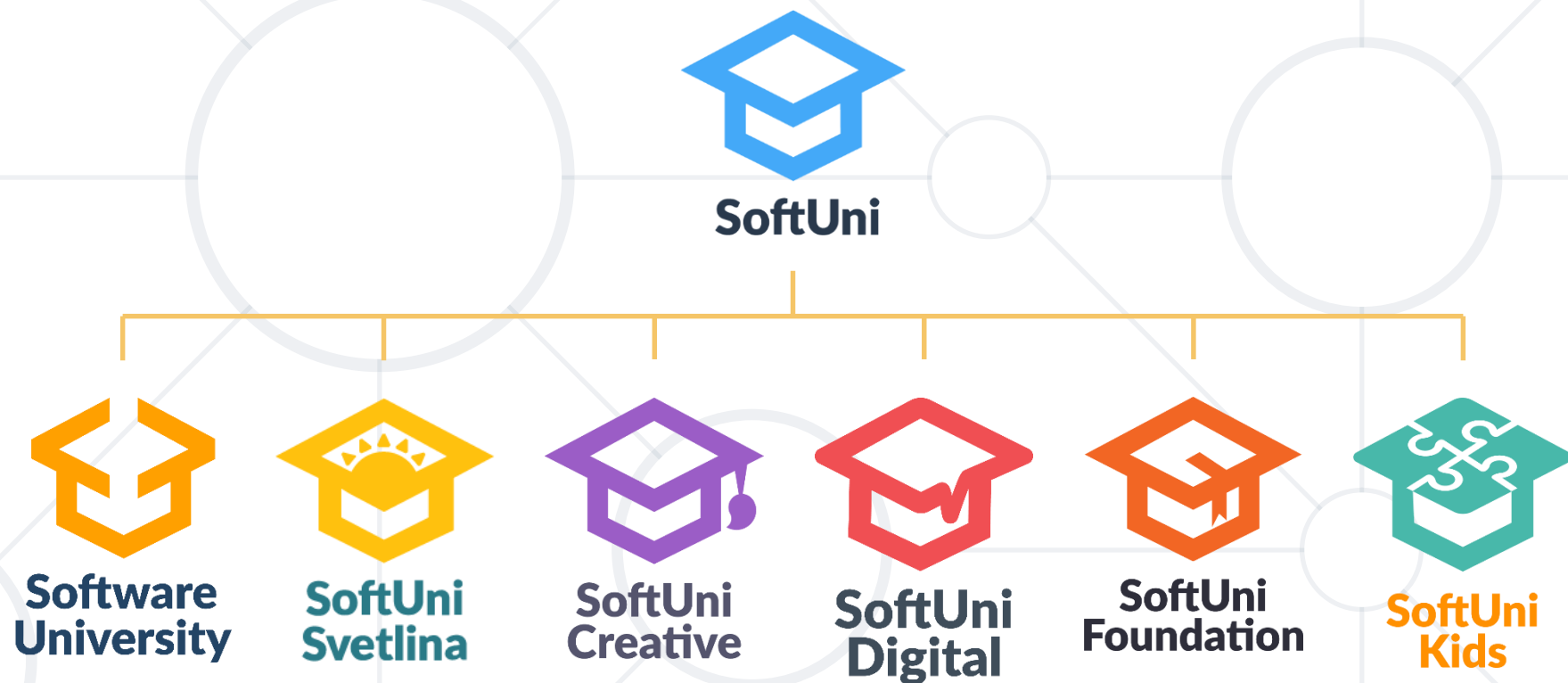
- На последният етаж винаги има големи апартаменти, които се означават с 'L', вместо с 'A'
- Ако има само един етаж, то има само големи апартаменти
- Примерен вход и изход:



```
int floors; cin >> floors;
int rooms; cin >> rooms;
for (int i = floors; i >= 1; i--) {
    for (int j = 0; j < rooms; j++) {
        if (i == floors)
            cout << "L" << i << j << " ";
        // TODO: print according to floor number
    }
    cout << endl;
}
```

Вложеният цикъл
итерира стаяте

Въпроси?



SoftUni Diamond Partners



XSsoftware



SBTech
we know sports



telenor



SoftwareGroup
doing it right

NETPEAK



SmartIT



Postbank

Решения за твоето утре



INDEAVR

Serving the high achievers



INFRAGISTICS®



STEMO®
Computer Systems & Software

SUPERHOSTING.BG

SoftUni Organizational Partners



OneE
SOFTWARE



WORLD
OF
MYTHS

- Настоящият курс (слайдове, примери, видео, задачи и др.) се разпространяват под свободен лиценз "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International"



- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg
- Software University Foundation
 - <http://softuni.foundation/>
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg

