

## **Лекція 8. Вкладені цикли.**

---

Цикли можуть бути вкладеними.

Наприклад, у цикл **for** можна вкласти інший цикл **for**.

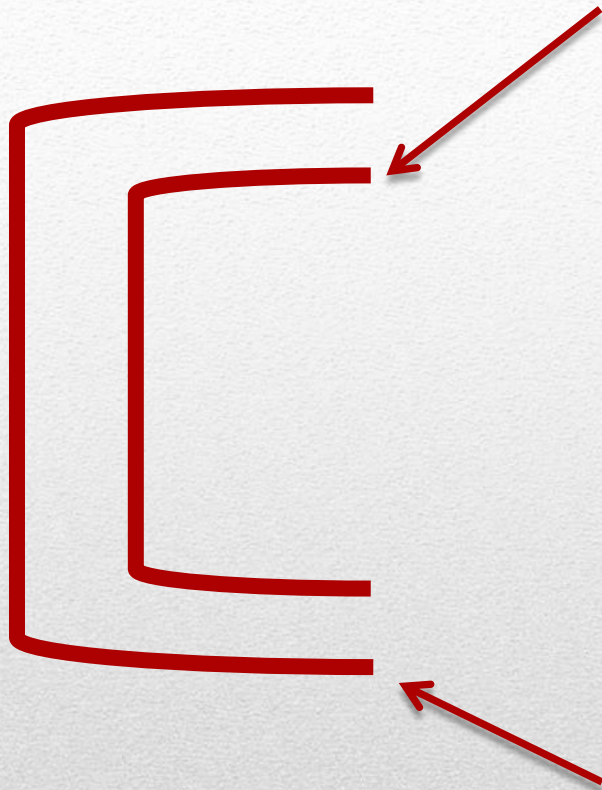
Або у цикл **for** вкласти цикл **while**.

Чи у **do ... while** вкласти цикл **for**.

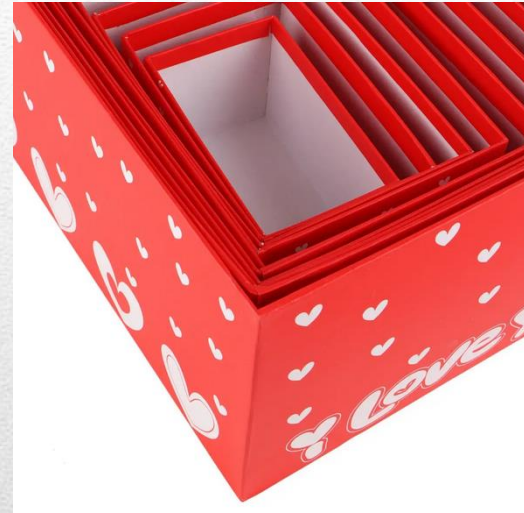
**Приклад задачі з вкладеними циклами:**

Написати програму, яка буде виводити таблицю множення чисел від 1 до 9.

---



**вкладений**



**зовнішній**

Вкладені цикли організовані таким чином: внутрішній цикл повністю вміщується в тілі зовнішнього циклу

---

Напишемо програму:

*зовнішній  
цикл*

```
int beginValue = 1,  
    endValue = 9;  
int i, j;  
for(i = beginValue; i <= endValue; i++)  
    for (j = beginValue; j <= endValue; j++)  
        printf("%d * %d == %d\n", i, j, i * j);
```

*внутрішній цикл*

E:\CI(Visual Stu

```
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
2 * 1 == 2
2 * 2 == 4
2 * 3 == 6
2 * 4 == 8
2 * 5 == 10
2 * 6 == 12
2 * 7 == 14
2 * 8 == 16
2 * 9 == 18
```

```
3 * 1 == 3
3 * 2 == 6
3 * 3 == 9
3 * 4 == 12
3 * 5 == 15
3 * 6 == 18
3 * 7 == 21
3 * 8 == 24
3 * 9 == 27
4 * 1 == 4
4 * 2 == 8
4 * 3 == 12
4 * 4 == 16
4 * 5 == 20
4 * 6 == 24
4 * 7 == 28
4 * 8 == 32
4 * 9 == 36
```

```
5 * 1 == 5
5 * 2 == 10
5 * 3 == 15
5 * 4 == 20
5 * 5 == 25
5 * 6 == 30
5 * 7 == 35
5 * 8 == 40
5 * 9 == 45
6 * 1 == 6
6 * 2 == 12
6 * 3 == 18
6 * 4 == 24
6 * 5 == 30
6 * 6 == 36
6 * 7 == 42
6 * 8 == 48
6 * 9 == 54
```

```
7 * 1 == 7
7 * 2 == 14
7 * 3 == 21
7 * 4 == 28
7 * 5 == 35
7 * 6 == 42
7 * 7 == 49
7 * 8 == 56
7 * 9 == 63
8 * 1 == 8
8 * 2 == 16
8 * 3 == 24
8 * 4 == 32
8 * 5 == 40
8 * 6 == 48
8 * 7 == 56
8 * 8 == 64
8 * 9 == 72
```

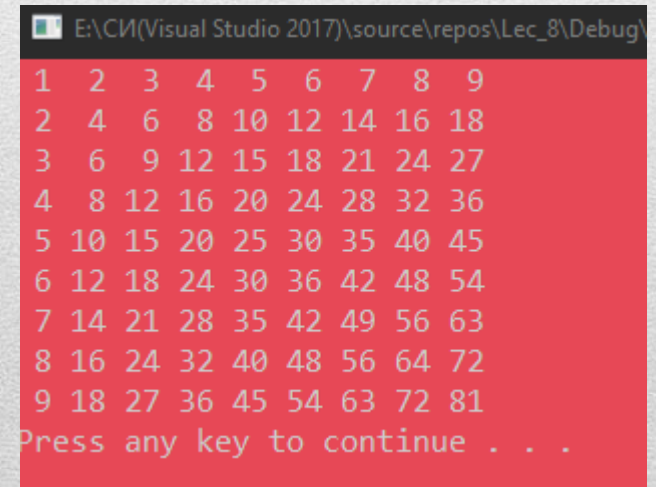
```
9 * 1 == 9
9 * 2 == 18
9 * 3 == 27
9 * 4 == 36
9 * 5 == 45
9 * 6 == 54
9 * 7 == 63
9 * 8 == 72
9 * 9 == 81
Press any key
```

# Написати програму, яка буде виводити таблицю МНОЖЕННЯ

```
#include <stdio.h>
#include<stdlib.h>

int main() {

    for (int a = 1; a < 10; a++) {
        for (int b = 1; b < 10; b++)
            printf("%2d ", a * b);
        printf("\n");
    }
    system("pause");
    return 0;
}
```



```
E:\CUI(Visual Studio 2017)\source\repos\Lec_8\Debug
1  2  3  4  5  6  7  8  9
2  4  6  8 10 12 14 16 18
3  6  9 12 15 18 21 24 27
4  8 12 16 20 24 28 32 36
5 10 15 20 25 30 35 40 45
6 12 18 24 30 36 42 48 54
7 14 21 28 35 42 49 56 63
8 16 24 32 40 48 56 64 72
9 18 27 36 45 54 63 72 81
Press any key to continue . . .
```

Розглянемо ще одне завдання:  
вивести на екран ось такий візерунок:

```
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
```

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include<stdlib.h>

int main() {
    int rowCount = 10, colCount = 12;
    int i, j;
    for (i = 0; i < rowCount; i++)
    {
        for (j = 0; j < colCount; j++)
            if ((i + j) % 2 == 0)
                printf("[");
            else
                printf("]");
        printf("\n");
    }

    system("pause");
    return 0;
}
```

---



```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include<stdlib.h>

int main() {
    int rowCount = 10, colCount = 12;
    int i, j;
    for (i = 0; i < rowCount; i++)
    {
        for (j = 0; j < colCount; j++)
            printf((i + j) % 2 ? "]" : "[");
        printf("\n");
    }

    system("pause");
    return 0;
}
```

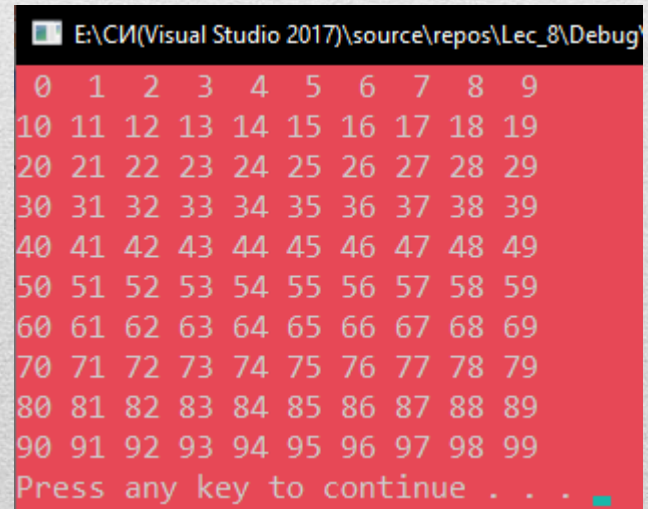
---

Вивести числа від 0 до 99, по 10 в кожному рядку

```
#include <stdio.h>
#include<stdlib.h>

int main() {
    for (int i = 0; i < 10; i++){
        for (int j = 0; j < 10; j++)
            printf("%2d ", i * 10 + j);
        printf("\n");
    }

    system("pause");
    return 0;
}
```



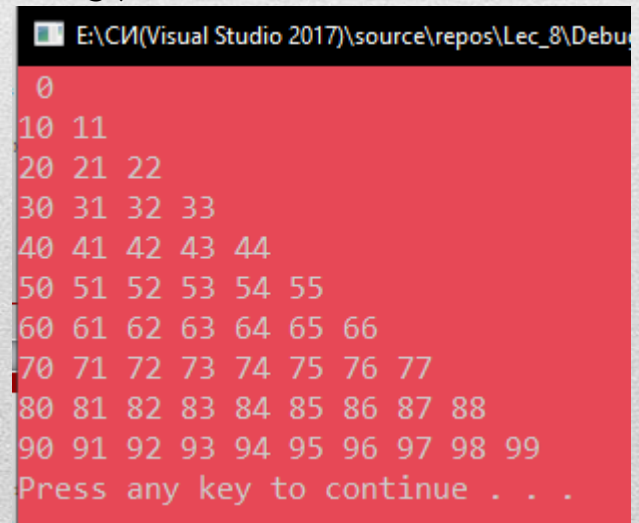
```
E:\СИ(Visual Studio 2017)\source\repos\Lec_8\Debug
 0  1  2  3  4  5  6  7  8  9
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55 56 57 58 59
60 61 62 63 64 65 66 67 68 69
70 71 72 73 74 75 76 77 78 79
80 81 82 83 84 85 86 87 88 89
90 91 92 93 94 95 96 97 98 99
Press any key to continue . . .
```

Вивести числа від 0 до 99 нижче головної діагоналі

```
#include <stdio.h>
#include<stdlib.h>

int main() {
    for (int i = 0; i < 10; i++){
        for (int j = 0; j < 10; j++){
            if (j > i) break;
            printf("%2d ", i * 10 + j);
        }
        printf("\n");
    }

    system("pause");
    return 0;
}
```




```
E:\СИ(Visual Studio 2017)\source\repos\Lec_8\Debug
0
10 11
20 21 22
30 31 32 33
40 41 42 43 44
50 51 52 53 54 55
60 61 62 63 64 65 66
70 71 72 73 74 75 76 77
80 81 82 83 84 85 86 87 88
90 91 92 93 94 95 96 97 98 99
Press any key to continue . . .
```

Можна робити і потрібні цикли.

Розташуємо два цикли всередині зовнішнього на 3 ітерації.

```
#include <stdio.h>
#include<stdlib.h>

int main() {
int k = 0;
while (k < 3) {
    for (int i = 1; i <= 5; i++) {
        for (int j = 0; j < i; j++)
            printf("*");
        printf("\n");
    }
    k++;
}
system("pause");
return 0;
}
```



```
E:\СИ(Visual Studio 2017)\sourc
*
**
***
****
*****
*
**
***
****
*****
*
**
***
****
*****
Press any key to conti
```

Дано натуральне число. Визначити кількість цифр 3 в ньому.

Наприклад: **12343**

```
int n;           // задане натуральне число
int k=0;        // кількість цифр 3 у числі
int t, d;       // додаткові змінні

n = 12343;

t = n;          // копія з n

while (t > 0)
{
    d = t % 10; // виділити останню цифру
    if (d == 3) k++;
    t = t / 10; // зменшити розрядність числа
}
printf("k=%d\n", k);
```

---

Дано натуральне число.

Наприклад: **12343**

Знайти суму цифр.

```
t = n;  
sum = 0;  
while (t > 0)  
{  
    sum += t % 10;  
    t = t / 10;  
}  
printf("sum=%d\n", sum);
```

---