

**Вкладені цикли.  
Покрокове введення та  
виведення даних**

Лекція №8

# ***Вкладені цикли***

Оскільки “тіло” циклу може містити будь-який оператор, будь-яку послідовність дій, то у якості такого оператору може бути використаний оператор циклу.

Таким чином можна конструювати **цикли в циклах** або ж **вкладені цикли**.

Цикл, який міститься у тілі іншого циклу, називається **вкладеним циклом**.

Вкладений цикл щодо циклу, в тіло якого він вкладений, називається **внутрішнім циклом**.

Цикл, в тілі якого існує вкладений цикл, називається **зовнішнім** щодо вкладеного.

Усередині вкладеного циклу може бути наступний вкладений цикл, утворюючи наступний **рівень вкладеності** і так далі.

Кількість рівнів вкладеності, як правило, не обмежується.

## Приклад 1.1. Сформулювати умову!

```
int i,j,n,m;  
printf("zadajte n, m: ");  
scanf("%d%d",&n,&m);  
for (i=1;i<=n;i++)  
    for (j=1;j<=m;j++)  
        printf("%d %d\n",i,j);
```

Чому у вкладених циклах  
параметри-змінні **різні**: *i* та *j*?

Результат: ?????? Для n=2, m=2:

```
1 1  
1 2  
2 1  
2 2
```

Як змінити програму, щоб виведення було у форматі:

1: 1 2

2: 1 2 ??????

## Приклад 1.2.

```
int i,j,n,m;
```

```
printf ("zadajte n, m: ");
```

```
scanf ("%d%d",&n,&m);
```

```
for (i=1;i<=n;i++)
```

```
{
```

```
    printf ("%d:_",i);
```

```
    for (j=1;j<=m;j++) printf("%d_",j);
```

```
    printf("\n");
```

```
}
```

Чому для зовнішнього циклу використані фігурні дужки {}?



## Приклад 2.1. “Таблиця множення $n*m$ ”.

```
int i,j,n,m;  
printf("zadajte n, m:_");  
scanf("%d%d",&n,&m);  
for (i=1;i<=n;i++)  
{  
    for (j=1;j<=m;j++) printf("%d_",i*j);  
    printf("\n");  
}
```

Результат для  $n=5$ ,  $m=3$ ????

1	2	3
2	4	6
3	6	9
4	8	12
5	10	15

## Приклад 2.2. “Таблиця множення $n*m$ ”

```
int i,j,n,m;  
printf("zadajte n, m:_");  
scanf("%d%d",&n,&m);  
for (i=1;i<=n;i++)  
{  
    for (j=1;j<=m;j++)  
        printf("%d\t",i*j);  
    printf("\n");  
}
```

Результат для  $n=5$ ,  $m=3$ ????

1	2	3
2	4	6
3	6	9
4	8	12
5	10	15

Приклад 3. Обчислити значення виразу  $\sum_{i=1}^n \sum_{j=1}^m (i + j)$ .

$[(1+1)+(1+2)+(1+3)+\dots+(1+m)] + [(2+1)+(2+2)+\dots+(2+m)] + \dots + [(n+1)+(n+2)+\dots+(n+m)]$

```
int i,j,n,m,sum=0;
```

```
printf("zadajte n, m: ");
```

```
scanf("%d%d",&n,&m);
```

```
for (i=1;i<=n;i++)
```

```
    for (j=1;j<=m;j++) sum+=i+j;
```

```
printf("%d",sum);
```

Результат для n=2, m=2 ????



**Приклад 4.** Обчислити значення виразу  $\sum_{i=1}^n \prod_{j=1}^m (i+j)$ .

$[(1+1)*(1+2)*(1+3)*\dots*(1+m)] + [(2+1)*(2+2)*\dots*(2+m)] + \dots + [(n+1)*\dots*(n+m)]$

```
int i,j,n,m,sum=0,mult;
printf("zadajte n, m:_");
scanf("%d%d",&n,&m);
for (i=1;i<=n;i++)
{
    mult=1;
    for (j=1;j<=m;j++) mult*=i+j;
    sum+=mult;
}
printf("%d",sum);
```

**mult=1;** // чому ініціалізація добутку тут?

**for (j=1;j<=m;j++) mult\*=i+j;**

**sum+=mult;**

}

**Приклад 5.** Дано натуральне число  $n$ .  
Визначити, чи можна представити це число у вигляді  $n = x^2 + y^2 + z^2$ , де  $x, y, z$  - натуральні числа.

```
flag=0;
```

```
for (x=1; x<=sqrt(n); x++)
```

```
    for (y=1; y<=sqrt(n)-x*x; y++)
```

```
        for (z=1; z<=sqrt(n)-x*x-y*y; z++)
```

```
            if (n==x*x+y*y+z*z)
```

```
            {
```

```
                printf("%d_%d_%d",x,y,z);
```

```
                flag=1;
```

```
            }
```

```
if (flag==0) printf("No");
```

Яке призначення змінної **flag**?

Чи потрібно умову **flag==1**  
перевіряти у циклах?

# ***Покрокове введення та виведення даних***

Існує цілий клас задач, які обробляють деяку послідовність уведених значень. Однак при цьому немає необхідності **одночасно зберігати у пам'яті комп'ютера** усі ці значення.

Такі алгоритми відносять до алгоритмів **покрокового введення** вхідних даних.

**Приклад 6.** Дано натуральне число  $n$  і дійсні числа  $a_1, a_2, \dots, a_n$ .  
Обчислити значення  $\sin(a_1 + a_2 + \dots + a_n) \cos(a_1 a_2 \dots a_n)$ .

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

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

```
#include <math.h>
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    int n,i; float a,sum=0,mult=1;
```

```
    printf("n="); scanf("%d",&n);
```

```
    for (i=1; i<=n;i++)
```

```
    {
```

```
        scanf("%f",&a);
```

```
        sum+=a; mult*=a;
```

```
    }
```

```
    printf("%.3f\n",sin(sum)*cos(mult));
```

```
    system("PAUSE");
```

```
    return 0;
```

```
}
```

**Приклад 7.** Дано натуральне число  $n$  і дійсні числа  $a_1, a_2, \dots, a_n$ . Обчислити  $\max(a_1, (a_1+a_2), (a_1+a_2+a_3), \dots, (a_1+a_2+\dots+a_n))$ .

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

```
int main(int argc, char *argv[])
{
    int n,i; float a,sum,max;
    printf("n="); scanf("%d",&n);
    printf("a="); scanf("%f",&max); // ЧОМУ max???
    sum=max;
    for (i=1; i<n;i++)
    {
        printf("a="); scanf("%f",&a);
        sum+=a;
        if (max<sum) max=sum;
    }
    printf("%.3f\n",max);
    system("PAUSE");
    return 0;
}
```



**Приклад 8.** Дано натуральне число  $n$  і дійсні числа  $a_1, a_2, \dots, a_n$ . Обчислити  $\max(a_1, a_2, \dots, a_n) + \min(a_1, a_2, \dots, a_n)$ .

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

```
int main(int argc, char *argv[])
{
    int n,i; float a,max, min;
    printf("n="); scanf("%d",&n);
    printf("a="); scanf("%f",&max);
    min=max;
    for (i=1; i<n;i++)
    {
        printf("a="); scanf("%f",&a);
        if (max<a) max=a;
        if (min>a) min=a;
    }
    printf("%.3f \n",max+min);
    system("PAUSE");
    return 0;
}
```

Якщо за умовою задачі є необхідність отримувати результати на **кожному** поточному кроці виконання циклу, то такі задачі відносять до задач **покрокового виведення** даних.

**Приклад 9.** Дано натуральне число  $n$ . Отримати послідовність значень  $b_i$  при  $i=1,2,\dots,n$ , якщо відомо, що  $b_i = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{i}$ .

```
int main(int argc, char *argv[])
{
    int n,i; float b;
    printf("n="); scanf("%d",&n);
    b=0;
    for (i=1; i<=n; i++)
    {
        b+=(float)1/i;
        printf("%.3f_",b);
    }
    system("PAUSE");
    return 0;
}
```

**Приклад 10.** Дано натуральне число  $n$  і дійсні числа  $a_1, a_2, \dots, a_n$ .  
Обчислити  $a_1, a_1+a_2, \dots, a_1+a_2+\dots+a_n$ .

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

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

```
int main(int argc, char *argv[])
```

```
{
```

```
int n,i; float a, sum=0;
```

```
printf("n="); scanf("%d",&n);
```

```
for (i=1; i<=n; i++)
```

```
{
```

```
scanf("%f",&a);
```

```
sum+=a;
```

```
printf("%.3f_",sum);
```

```
}
```

```
printf("\n");
```

```
system("PAUSE");
```

```
return 0;
```

```
}
```

**Приклад 11.** Дано натуральне число  $n$  і дійсні числа  $a_1, a_2, \dots, a_n$ .  
Обчислити  $a_1, a_1 * a_2, \dots, a_1 * a_2 * \dots * a_n$ .

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

```
int main(int argc, char *argv[])
{
    int n,i; float a,mult=1;
    printf("n="); scanf("%d",&n);
    for (i=1; i<=n;i++)
    {
        scanf("%f",&a);
        mult*=a;
        printf("%.3f_",mult);
    }
    printf("\n");
    system("PAUSE");
    return 0;
}
```



Подекуди при складанні циклічних алгоритмів зручно використовувати так звані „**перемикачі**”.

**1.  $k = -k$ .**

При початковому значенні  $k=1$  отримуватимемо послідовність  $-1, 1, -1, 1, \dots$

При початковому значенні  $k = -1$  ця послідовність матиме вигляд  $1, -1, 1, -1, \dots$

**Приклад 12.** Дано натуральне число  $n$ . Отримати послідовність значень  $b_i$  при  $i=1,2,\dots,n$ , якщо відомо, що  $b_i = i(1 - 2 + 3 - \dots + (-1)^{i+1}i)$ .

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

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

```
int main(int argc, char *argv[])
```

```
{
```

```
    int n,i,b,k=1,sum=0;
```

```
    printf("n="); scanf("%d",&n);
```

```
    for (i=1; i<=n;i++)
```

```
    {
```

```
        sum+=k*i;
```

```
        k=-k;
```

```
        b=i*sum; // printf("%d\n", i*sum);
```

```
        printf("%d\n",b);
```

```
    }
```

```
    system("PAUSE");
```

```
    return 0;
```

```
}
```

## 2. $k = 1-k$ .

При початковому значенні  $k=1$   
отримуємо послідовність  $0, 1, 0, 1, \dots$

При початковому значенні  $k = 0$  ця  
послідовність матиме вигляд  $1, 0, 1, 0, \dots$

**Приклад 13.** Дано натуральне число  $n$ . Отримати послідовність значень  $b_i$  при  $i=1,2,\dots,n$ , якщо відомо, що  $b_i = 5(1 + 3 + \dots + (2i - 1))$ .

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

```
int main(int argc, char *argv[])
{
    int n,i,b,k=1,sum=0;
    printf("n="); scanf("%d",&n);
    for (i=1; i<=2*n; i++)
    {
        sum+=k*i;
        k=1-k;
        b=5*sum; // printf("%d\n", 5*sum);
        printf("%d\n",b);
    }
    system("PAUSE");
    return 0;
}
```

Альтернативний варіант:

```
for (i=1; i<=2*n-1;i+=2)
{
    sum+=i;
    printf("%d\n", 5*sum);
}
```

# Рекурентні послідовності

У математиці серед “золотих” чисел почесне місце посіли числа Фібоначчі (Леонардо Пізанський, XIIIст.). Кажуть, що свого часу відомий вчений в послідовності цих чисел відобразив закон розмноження кроликів. Ці числа мають такий вигляд:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ... .

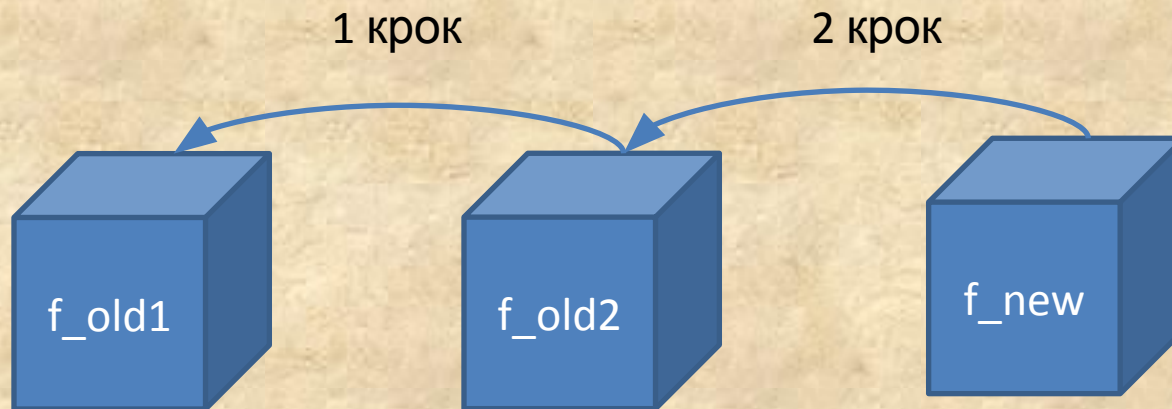
Можна помітити, що починаючи з третього числа, кожне наступне дорівнює сумі двох попередніх. Тобто, починаючи з третього члена цієї послідовності існує така залежність:

$$F_{\text{new}} = F_{\text{old1}} + F_{\text{old2}} .$$

Залежність нового значення елемента послідовності від певної кількості попередніх називається **рекурентною**, а програми, які використовують рекурентні формули, називаються відповідно **рекурентними**.



# ***Схематичне зображення рекурентності***



**Приклад 14.** За даними співвідношенням визначити  $n$ -ий елемент числової послідовності ( $n > 2$ ):

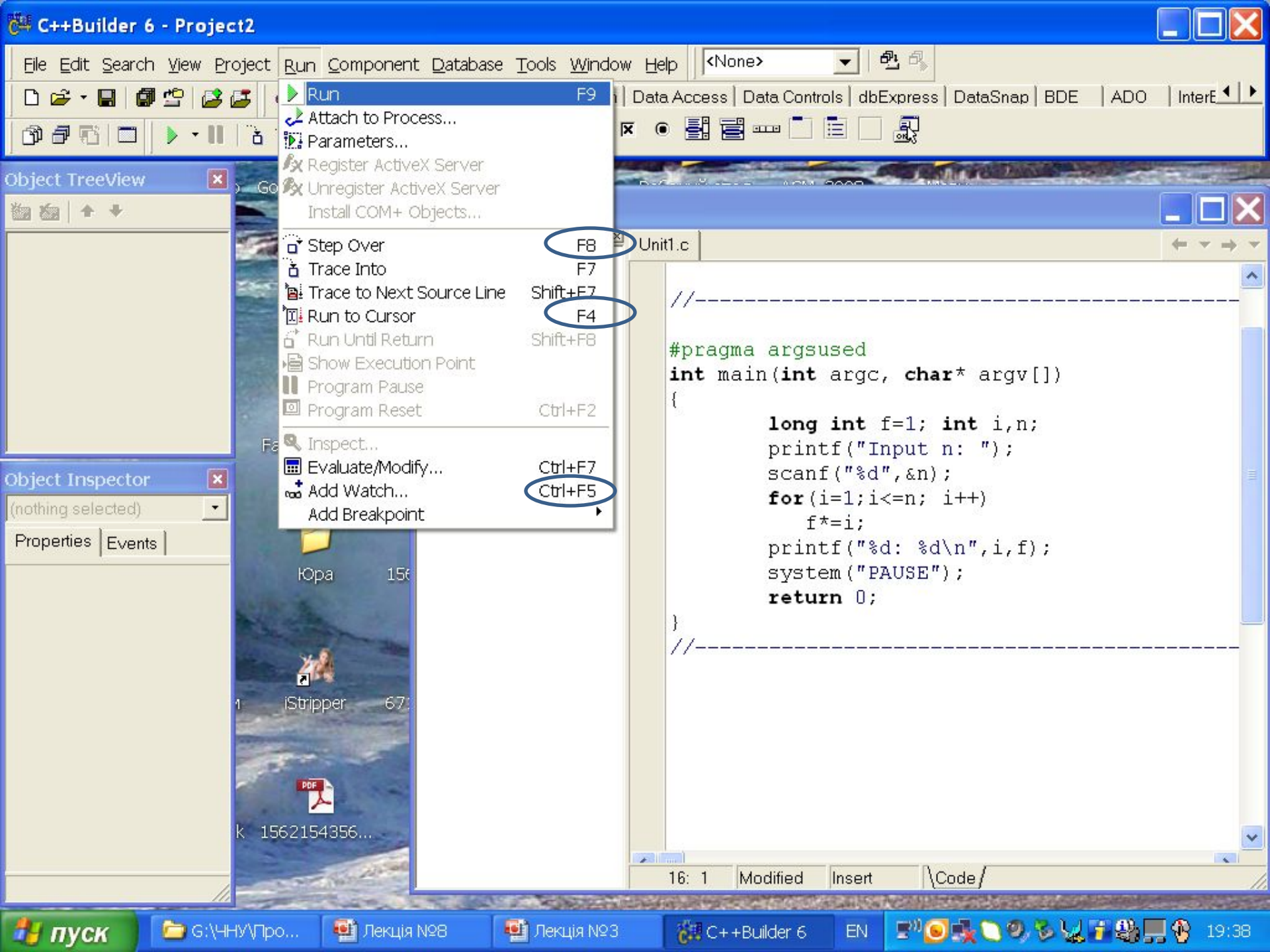
$$x_n = x_{n-1} + x_{n-2} + x_{n-3}; \quad x_0 = x_1 = 1, \quad x_2 = 6.$$

```
int n, i, x_0=1, x_1=1, x_2=6, x_new;
printf("n="); scanf("%d",&n);
if ((n==1) || (n==2)) printf("1");
else if (n==3) printf("6");
    else
    {
        for (i=4; i<=n; i++)
        {
            x_new=x_0+x_1+x_2;
            x_0=x_1; x_1=x_2; x_2=x_new;
        }
        printf("%d_",x_new);
    }
printf("\n");
```

**Приклад 15.** За даними співвідношенням визначити  $n$ -ий елемент числової послідовності ( $n > 2$ ):

$$x_n = x_{n-1} + 4x_{n-3}; \quad x_0 = x_1 = x_2 = 2.$$

```
int n, i, x_0, x_1, x_2, x_new;
printf("n="); scanf("%d",&n);
x_0=x_1=x_2=2;
if ((n==1) || (n==2) || (n==3)) printf("2");
else
{
    for (i=4; i<=n;i++)
    {
        x_new=4*x_0+x_2;
        x_0=x_1; x_1=x_2; x_2=x_new;
    }
    printf("%d\n",x_new);
}
```



- Run F9
- Attach to Process...
- Parameters...
- Register ActiveX Server
- Unregister ActiveX Server
- Install COM+ Objects...
- Step Over F8
- Trace Into F7
- Trace to Next Source Line Shift+F7
- Run to Cursor F4
- Run Until Return Shift+F8
- Show Execution Point
- Program Pause
- Program Reset Ctrl+F2
- Inspect...
- Evaluate/Modify... Ctrl+F7
- Add Watch... Ctrl+F5
- Add Breakpoint

```
Unit1.c  
//-----  
#pragma argsused  
int main(int argc, char* argv[])  
{  
    long int f=1; int i,n;  
    printf("Input n: ");  
    scanf("%d",&n);  
    for (i=1;i<=n; i++)  
        f*=i;  
    printf("%d: %d\n",i,f);  
    system("PAUSE");  
    return 0;  
}  
//-----  
16: 1 Modified Insert \Code/
```



C++Builder 6 - Project2

File Edit Search View Project Run Component Database Tools Window Help

Standard Additional Win32 System Data Access Data Controls dbExpress DataSnap BDE ADO Interf

Standard toolbar icons: File, Edit, Search, View, Project, Run, Component, Database, Tools, Window, Help, Standard, Additional, Win32, System, Data Access, Data Controls, dbExpress, DataSnap, BDE, ADO, Interf

```
Unit1.c  
Project2 - Classes  
Unit1.c  
#pragma hdrstop  
//-----  
#pragma argsused  
int main(int argc, char* argv[])  
{  
    long int f=1; int i,n;  
    printf("Input n: ");  
    scanf("%d",&n);  
    for(i=1;i<=n;i++)  
        f*=i;  
    printf("%d\n",f);  
    system("PAUSE");  
    return 0;  
}
```

10: 30 Modified Insert \Code/

C++ Warning] Unit1.c(11): W8065 Call to function 'printf' with no prototype  
C++ Warning] Unit1.c(12): W8065 Call to function 'scanf' with no prototype  
C++ Warning] Unit1.c(15): W8065 Call to function 'printf' with no prototype

Watch List

i: [process not accessible]
n: [process not accessible]

Watches

Taskbar: 1562154356... лілія Дејкстра

Taskbar: Корзина

Taskbar: пуск G:\ЧНУ\Про... C++Builder 6 Лекція №8 Лекція №3 EN 18:40



```

Unit1.c
Project2 - Classes
Unit1.c

#pragma hdrstop

//-----

#pragma argsused
• int main(int argc, char* argv[])
{
•     long int f=1; int i,n;
•     printf("Input n: ");
•     scanf("%d",&n);
•     for(i=1;i<=n;i++)
•         f*=i;
•     printf("%d\n",f);
•     system("PAUSE");
•     return 0;
• }

13: 1 Modified Insert \Code/

```

C++ Warning] Unit1.c(11): W8065 Call to function 'printf' with no prototype

C++ Warning] Unit1.c(12): W8065 Call to function 'scanf' with no prototype

C++ Warning] Unit1.c(15): W8065 Call to function 'printf' with no prototype

время ООБЪЕКТ

Текст Клипы мульт...

4 6 8 10 12

Watch List

i: 256

n: 7

f: 1

Watches

Выбрать C:\Program Files\B...

Input n: 7

```
Unit1.c  
Project2 - Classes  
Unit1.c  
#pragma hdrstop  
//-----  
#pragma argsused  
int main(int argc, char* argv[])  
{  
    long int f=1; int i,n;  
    printf("Input n: ");  
    scanf("%d",&n);  
    for(i=1;i<=n; i++)  
        f*=i;  
    printf("%d\n",f);  
    system("PAUSE");  
    return 0;  
}
```

14: 1 Modified Insert \Code/

C++ Warning] Unit1.c(11): W8065 Call to function 'printf' with no prototype  
C++ Warning] Unit1.c(12): W8065 Call to function 'scanf' with no prototype  
C++ Warning] Unit1.c(15): W8065 Call to function 'printf' with no prototype

время Объект  
Текст  
Клипы мульт...

14: 1 6: 1 8: 1 10: 1 12: 1

Watch List

- i: 6
- n: 7
- f: 120

Watches

Background image of a slide with a textured, parchment-like appearance.



C++Builder 6 - Project2 [Stopped]

File Edit Search View Project Run Component Database Tools Window Help

Standard Additional Win32 System Data Access Data Controls dbExpress DataSnap BDE ADO InterF

Свернуть

Unit1.c

Project2 - Classes

```
#pragma hdrstop

//-----

#pragma argsused
• int main(int argc, char* argv[])
{
•     long int f=1; int i,n;
•     printf("Input n: ");
•     scanf("%d",&n);
•     for(i=1;i<=n;i++)
•         f*=i;
•     printf("%d\n",f);
•     system("PAUSE");
•     return 0;
• }

15: 1 Modified Insert \Code/
```

C++ Warning] Unit1.c(11): W8065 Call to function 'printf' with no prototype  
C++ Warning] Unit1.c(12): W8065 Call to function 'scanf' with no prototype  
C++ Warning] Unit1.c(15): W8065 Call to function 'printf' with no prototype

Watch List

i: 8
n: 7
f: 5040

Watches

Выбрать C:\Program Files\B...

Input n: 7

ДЕКЛАРАЦІЯ (29) Facebook 1562154356... ліля Дејкстра Корзина

пук G:\ЧНУ\... C++Buil... Лекція ... Лекція ... Выбрат... EN 18:57





C++Builder 6 - Project2 [Running]

File Edit Search View Project Run Component Database Tools Window Help

Standard Additional Win32 System Data Access Data Controls dbExpress DataSnap BDE ADO Interf

Standard toolbar icons: File, Edit, Search, View, Project, Run, Component, Database, Tools, Window, Help, and various utility icons.

Unit1.c

```
#pragma hdrstop

//-----

#pragma argsused
int main(int argc, char* argv[])
{
    long int f=1; int i,n;
    printf("Input n: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
        f*=i;
    printf("%d\n",f);
    system("PAUSE");
    return 0;
}
```

16: 1 Modified Insert \Code/

C++ Warning] Unit1.c(11): W8065 Call to function 'printf' with no prototype  
C++ Warning] Unit1.c(12): W8065 Call to function 'scanf' with no prototype  
C++ Warning] Unit1.c(15): W8065 Call to function 'printf' with no prototype

Watch List

- i: [process not accessible]
- n: [process not accessible]
- f: [process not accessible]

Watches

C:\Program Files\Borland\C...

```
Input n: 7
5040
Для продолжения нажмите любую к...
```







```

Unit1.c
Project2 - Classes
Unit1.c
• int main(int argc, char* argv[])
  {
  •     long int f=1; int i,n;
  •     printf("Input n: ");
  •     scanf("%d", &n);
  •     for(i=1;i<=n; i++)
  •         f*=i;
  •     printf("%d\n", f);
  •     system("PAUSE");
  •     return 0;
  • }
  //-----
  16: 1 Modified Insert \Code/
  [C++ Warning] Unit1.c(11): W8065 Call to function 'printf' with no prototype
  [C++ Warning] Unit1.c(12): W8065 Call to function 'scanf' with no prototype
  [C++ Warning] Unit1.c(15): W8065 Call to function 'printf' with no prototype
  Build/
  
```

Watch List

- i: [process not accessible]
- n: [process not accessible]
- f: [process not accessible]

Watches

C:\Program Files\Borland\CBuild...

```

Input n: 11
39916800
Для продовження натисніть будь-яку клавішу
  
```

```

Unit1.c
Project2 - Classes
Unit1.c
• int main(int argc, char* argv[])
  {
  •     long int f=1; int i,n;
  •     printf("Input n: ");
  •     scanf("%d", &n);
  •     for(i=1;i<=n; i++)
  •         f*=i;
  •     printf("%d\n", f);
  •     system("PAUSE");
  •     return 0;
  • }
  //-----
  16: 1 Modified Insert \Code/
  [C++ Warning] Unit1.c(11): W8065 Call to function 'printf' with no prototype
  [C++ Warning] Unit1.c(12): W8065 Call to function 'scanf' with no prototype
  [C++ Warning] Unit1.c(15): W8065 Call to function 'printf' with no prototype
  \Build/
  
```

Watch List

- i: [process not accessible]
- n: [process not accessible]
- f: [process not accessible]

Watches

C:\Program Files\Borland\CBuild...

```

Input n: 12
479001600
Для продолжения нажмите любую клавишу
  
```



C++Builder 6 - Project2 [Stopped]

File Edit Search View Project Run Component Database Tools Window Help <None>

Standard Additional Win32 System Data Access Data Controls dbExpress DataSnap BDE ADO Interf

Mouse Print Run A abt OK X

Canon LASER chromesetup Google Chrome Панель Барселона Рабочий стол ACM\_2008 Мови програмува...

```
Unit1.c  
Project2 - Classes  
Unit1.c  
//-----  
  
#pragma hdrstop  
  
//-----  
  
#pragma argsused  
• int main(int argc, char* argv[])  
{  
•     long int f=1; int i,n;  
•     printf("Input n: ");  
•     scanf("%d", &n);  
•     for(i=1;i<=n; i++)  
•         f*=i;  
•     printf("%d\n",f);  
•     system("PAUSE");  
•     return 0;  
• }  
//-----  
16: 1 Modified Insert \Code/
```

Watch List

i: 14
n: 13
f: 1932053504

Watches

C:\Program Files\Borland\CBuild...

```
Input n: 13  
1932053504
```

```
Unit1.c  
Project2 - Classes  
Unit1.c  
//-----  
#pragma argsused  
• int main(int argc, char* argv[])  
{  
•     long int f=1; int i,n;  
•     printf("Input n: ");  
•     scanf("%d",&n);  
•     for(i=1;i<=n; i++)  
•     {  
•         f*=i;  
•         printf("%d: %d\n",i,f);  
•     }  
•     system("PAUSE");  
•     return 0;  
• }  
//-----  
18: 1 Modified Insert \Code/
```

Watch List

- i: [process not accessible]
- n: [process not accessible]
- f: [process not accessible]

Watches

C:\Program Files\Borland\CBu...

```
Input n: 15  
1: 1  
2: 2  
3: 6  
4: 24  
5: 120  
6: 720  
7: 5040  
8: 40320  
9: 362880  
10: 3628800  
11: 39916800  
12: 479001600  
13: 1932053504  
14: 1278945280  
15: 2004310016  
Для продолжения нажмите любую кп
```