



Язык С

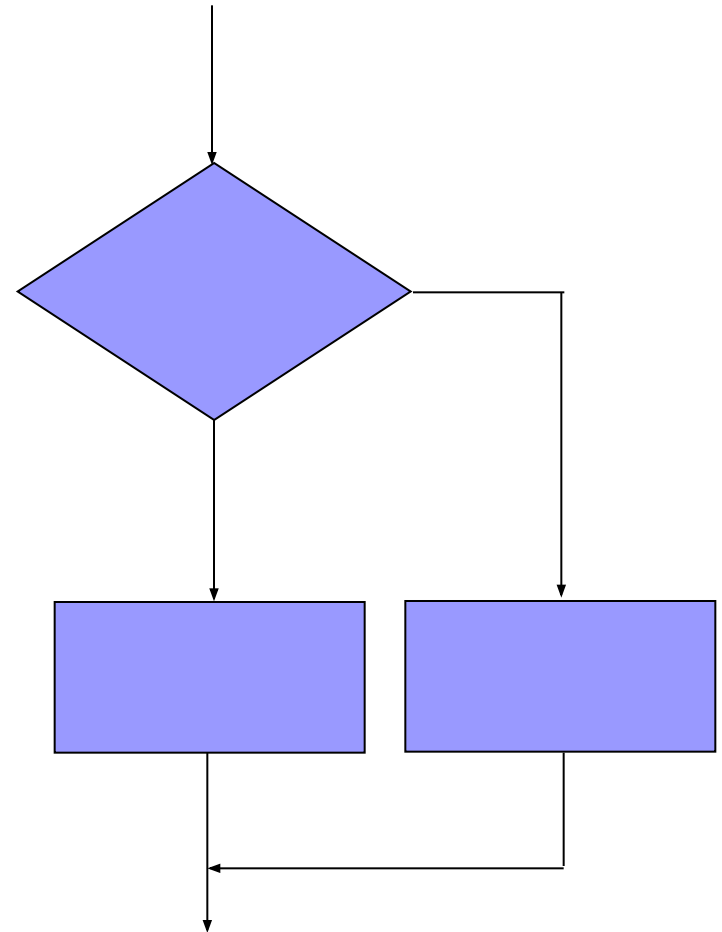
Лекция №2

Алгоритмические структуры

Условный оператор

```
if (условие)  
    оператор1;  
else        оператор2;
```

```
if (условие)  
    { оператор1;  
      оператор2;  
      ...  
    }  
else  
    { оператор3;  
      оператор4;  
      ...  
    }
```



Логические условия

- В языке C – целочисленная переменная (0 – ложь, не 0 – истина)
- В языке C++ есть тип **bool**, его значения **true** и **false**

Операции сравнения

`==` равно

`!=` не равно

`<` меньше

`<=` меньше, либо равно

`>=` больше, либо равно

`>` больше

```
if (x>0) y= 1;
```

```
else      y=-1;
```

Логические операции

- В математике $0 < x < 1$,
в языке C/C++ так нельзя!

- В языке C/C++ пишут

```
if ( (0 < x) && (x < 1) ) ...
```

Логические операции

! не

|| или (дизъюнкция)

&& и (конъюнкция)

Таблицы истинности

A \ B	T	F
T	T	F
F	F	F

И &&

A \ B	T	F
T	T	T
F	T	F

ИЛИ ||

A	!A
T	F
F	T

НЕ !

$$\neg (A \&\& B) \equiv (\neg A \parallel \neg B)$$

$$\neg (A \parallel B) \equiv (\neg A \&\& \neg B)$$

законы де Моргана

Виды оператора `if`

```
if    (x == 0)
    { оператор;
      оператор;
    }
```

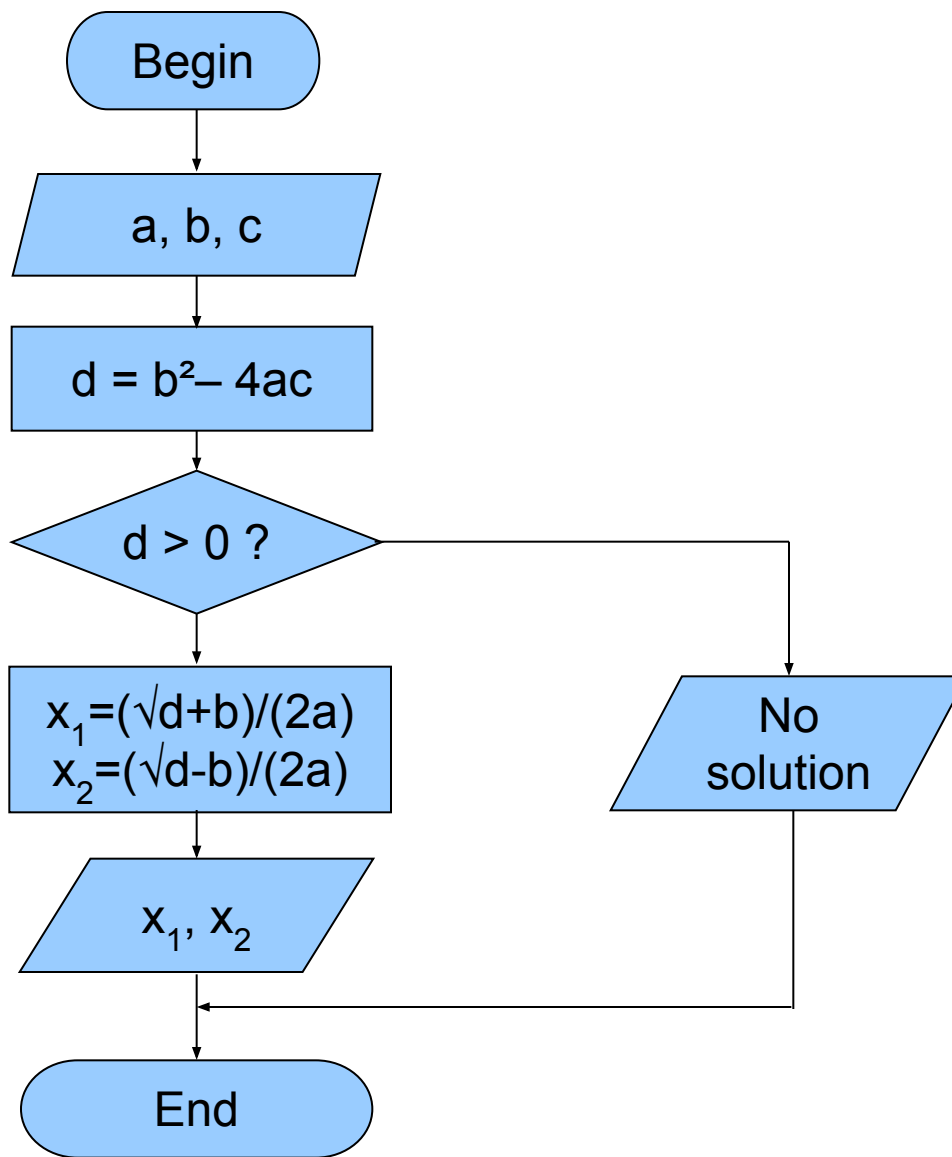
```
if (x>0) оператор;
else оператор;
```

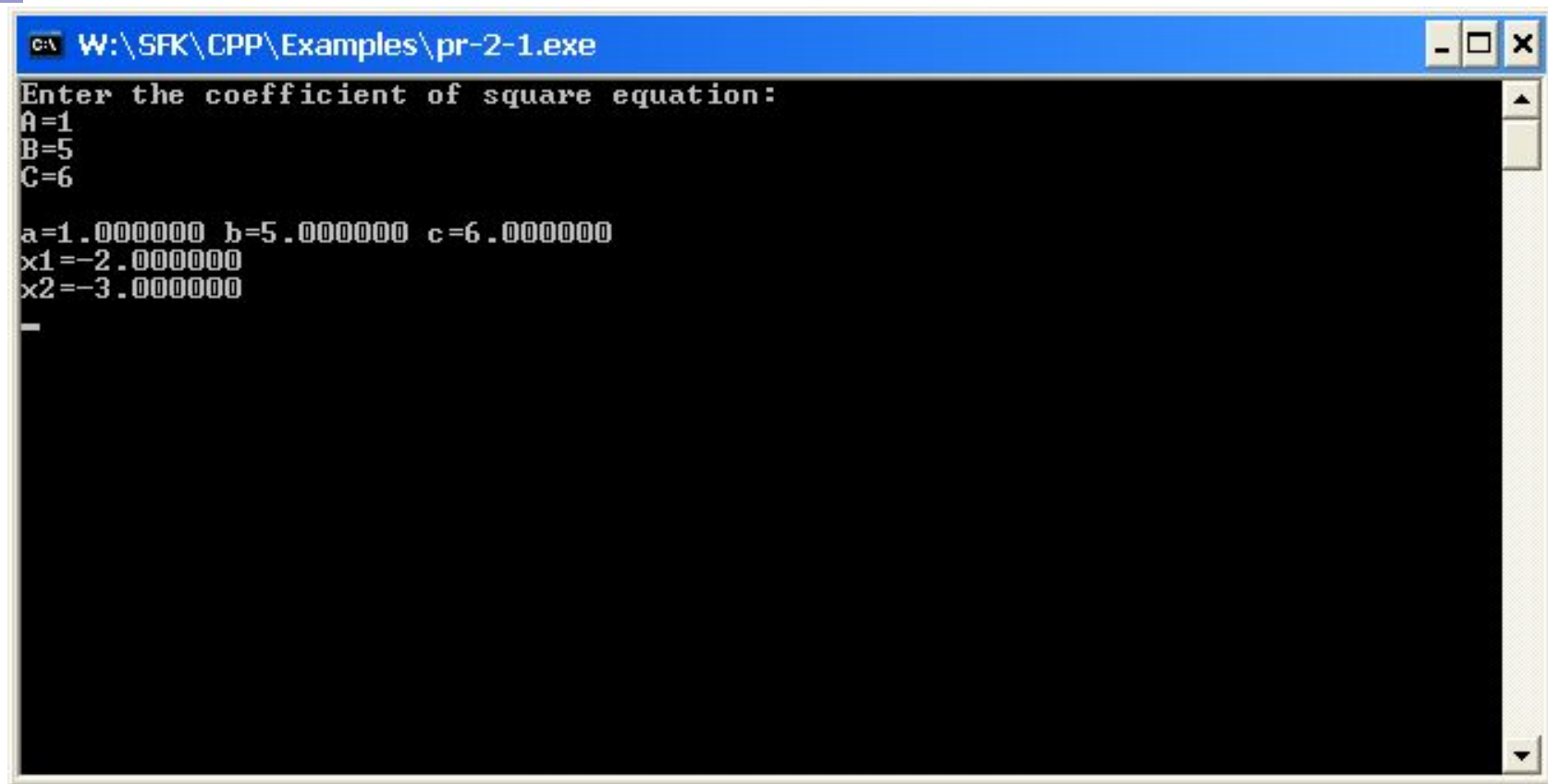
```
if (x == 0)
    { оператор;
      оператор;
    }
else
    { оператор;
      оператор;
    }
```


Вложенные операторы **if**

```
if (x<0) { z=-1; }  
else if (x==0) { z=0; }  
           else { z=1; }
```

Пример: решение квадратного уравнения





```
C:\W:\SFK\CPP\Examples\pr-2-1.exe
Enter the coefficient of square equation:
A=1
B=5
C=6

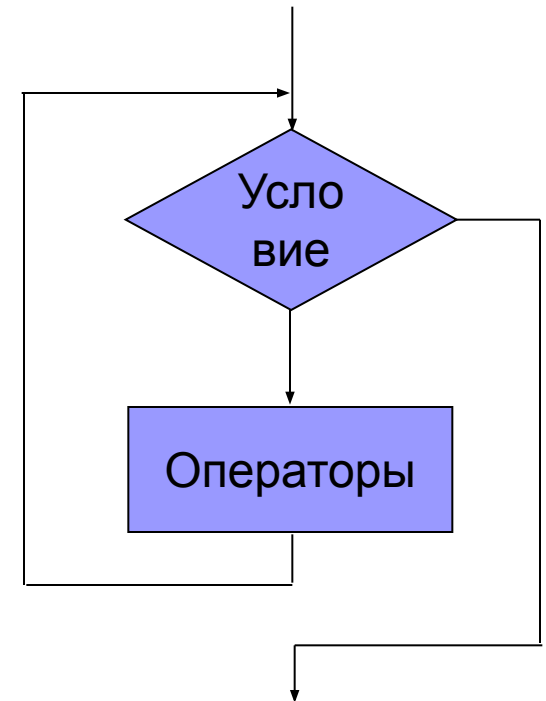
a=1.000000 b=5.000000 c=6.000000
x1=-2.000000
x2=-3.000000
-
```

```
else if (a==0) { x1= b/(2.0*a);  
    printf("x=%f\n",x1);  
}  
else { d=sqrt(d); a=2.0*a;  
    x1=(-b+d)/a; x2=(-b-d)/a;  
    printf("x1=%f\nx2=%f\n",x1,x2);  
}  
}
```

Цикл с предусловием

while (условие) *оператор;*

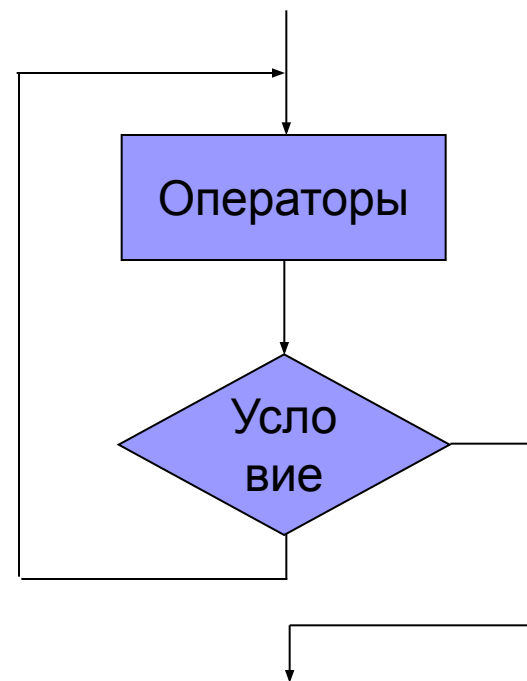
```
while (условие)
{
    оператор;
    оператор;
    оператор;
}
```



Цикл с постусловием

do *оператор* **while** (условие) ;

```
do
{
    оператор;
    оператор;
    оператор;
} while (условие) ;
```



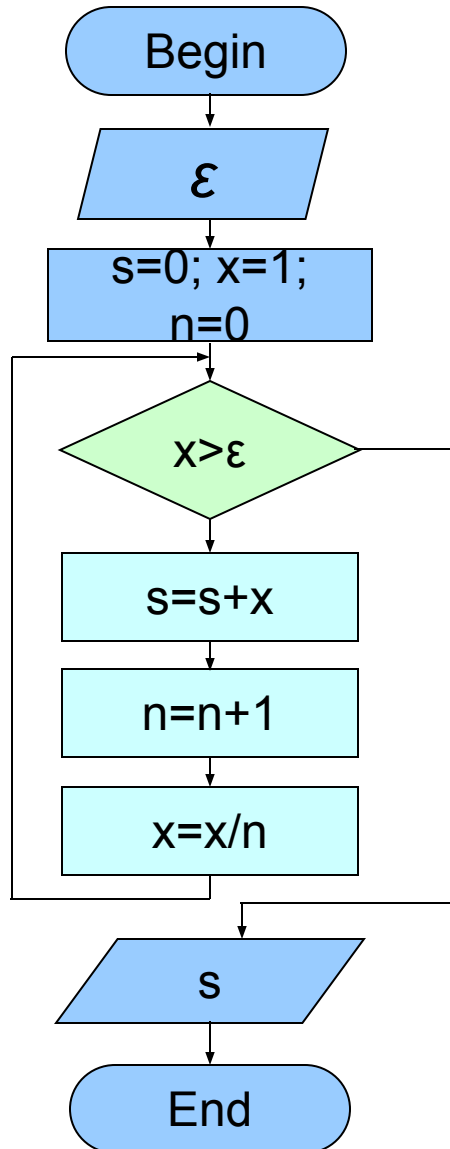
Вычисление числа

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = 1 + 1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{24} + \dots$$

s – накапливаемая сумма

x – член ряда

ε – требуемая точность



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

```
main () // Программа вычисления числа e
```

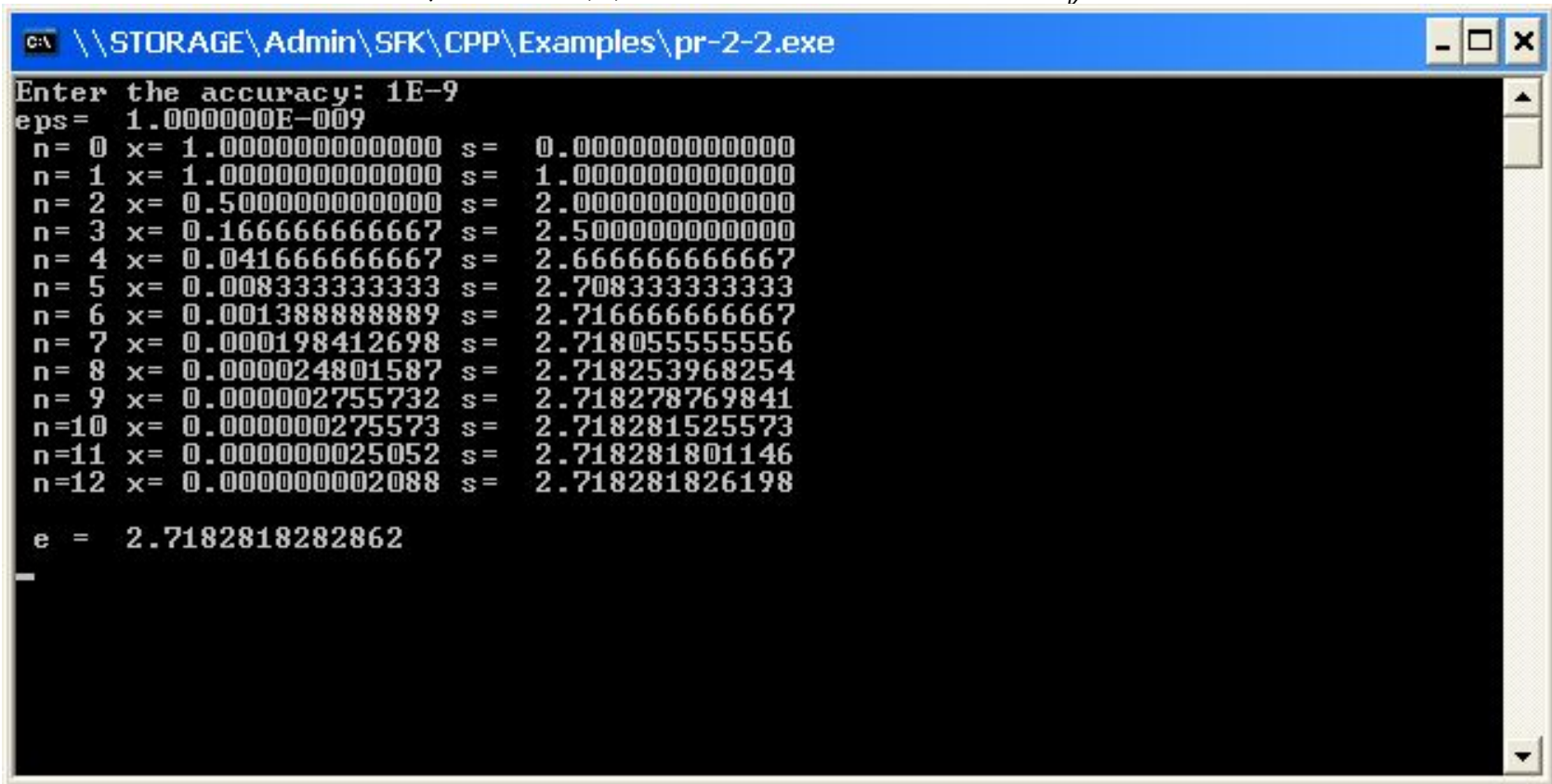
```
{
```

```
    double eps;    // Точность вычисления
```

```
    double x = 1.0; // Член ряда
```

```
    double s = 0;    // Накапливаемая сумма
```

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = 1 + 1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{24} + \dots$$



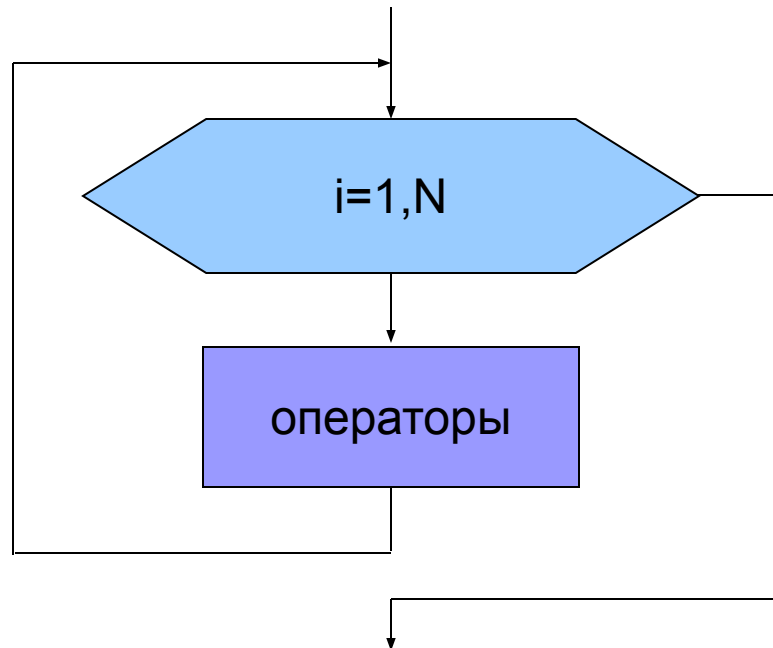
```
c:\ \STORAGE\Admin\SF\CPP\Examples\pr-2-2.exe
Enter the accuracy: 1E-9
eps= 1.000000E-009
n= 0 x= 1.00000000000000 s= 0.00000000000000
n= 1 x= 1.00000000000000 s= 1.00000000000000
n= 2 x= 0.50000000000000 s= 2.00000000000000
n= 3 x= 0.16666666666667 s= 2.50000000000000
n= 4 x= 0.04166666666667 s= 2.66666666666667
n= 5 x= 0.00833333333333 s= 2.70833333333333
n= 6 x= 0.00138888888889 s= 2.71666666666667
n= 7 x= 0.000198412698 s= 2.71805555555556
n= 8 x= 0.000024801587 s= 2.718253968254
n= 9 x= 0.000002755732 s= 2.718278769841
n=10 x= 0.000000275573 s= 2.718281525573
n=11 x= 0.000000025052 s= 2.718281801146
n=12 x= 0.000000002088 s= 2.718281826198

e = 2.7182818282862
```

Оператор цикла for

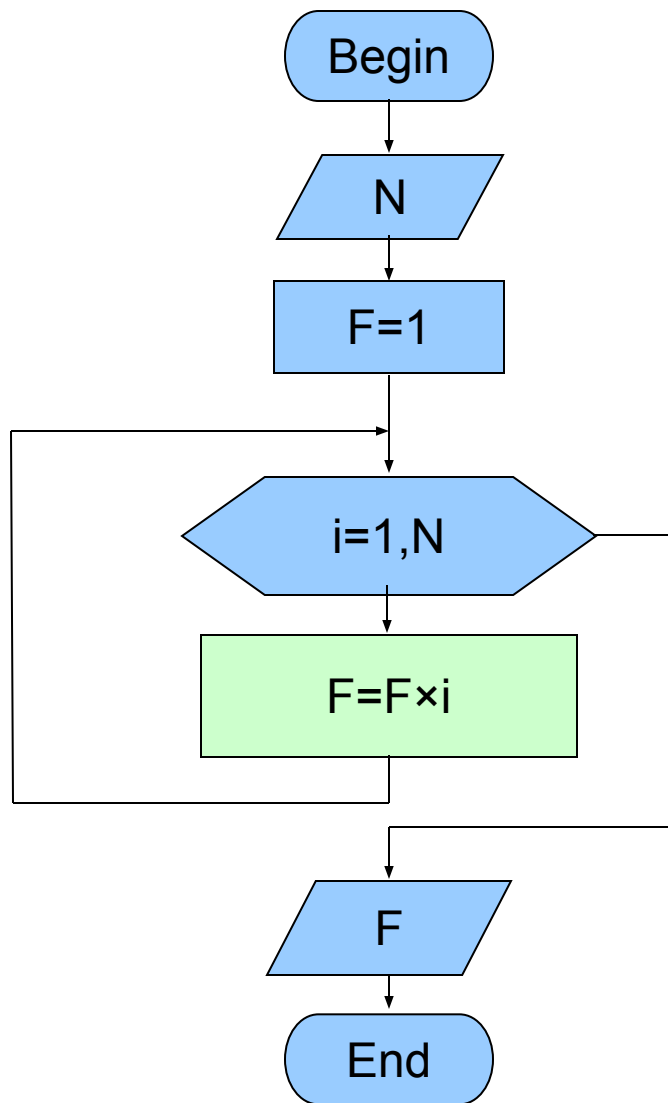
for (*инициализация; проверка условия; изменение*)
оператор;

```
for (i=0; i<10; i++)  
    printf("%d", i);
```



Вычисление факториала

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$$



Вычисление факториала

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

```
main () // Программа вычисления числа факториала
{
    int n;          // аргумент
    double f = 1;   // факториал
    printf("Enter n: "); scanf("%d", &n);
    for (int i = 1; i <= n; i++) f *= i;
    printf("\n %d! = %16.0lf \n", n, f);
}
```

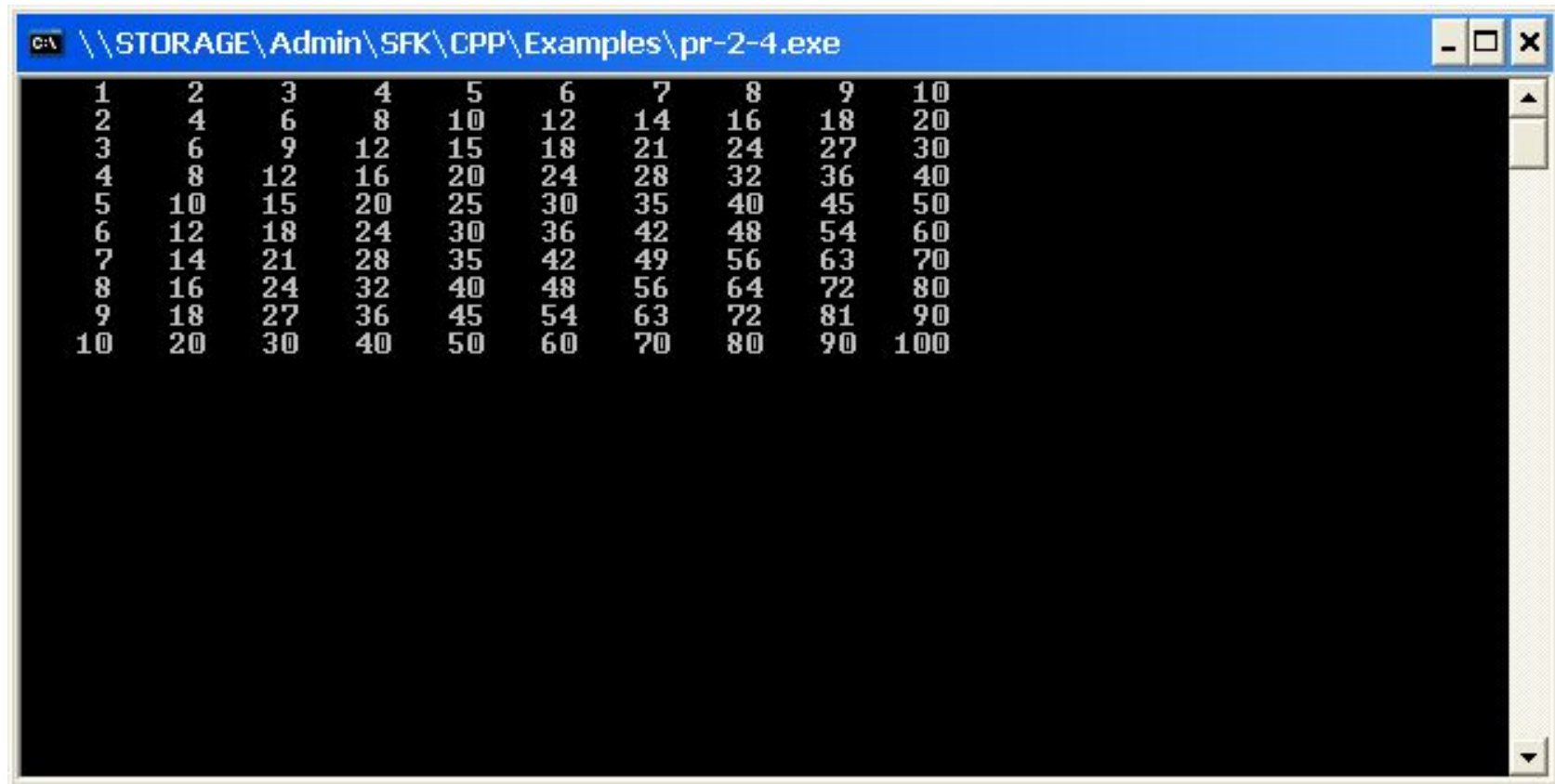
Вложенные циклы

```
#include <stdio.h>
#define N 10

main () // Таблица умножения
{
    int i,j; // аргумент

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

Вложенные циклы



A screenshot of a Windows command prompt window. The title bar shows the path `C:\ \STORAGE\Admin\SFk\CPP\Examples\pr-2-4.exe`. The window contains a 10x10 grid of numbers, where each row represents an iteration of an outer loop and each column represents an iteration of an inner loop. The numbers are arranged as follows:

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

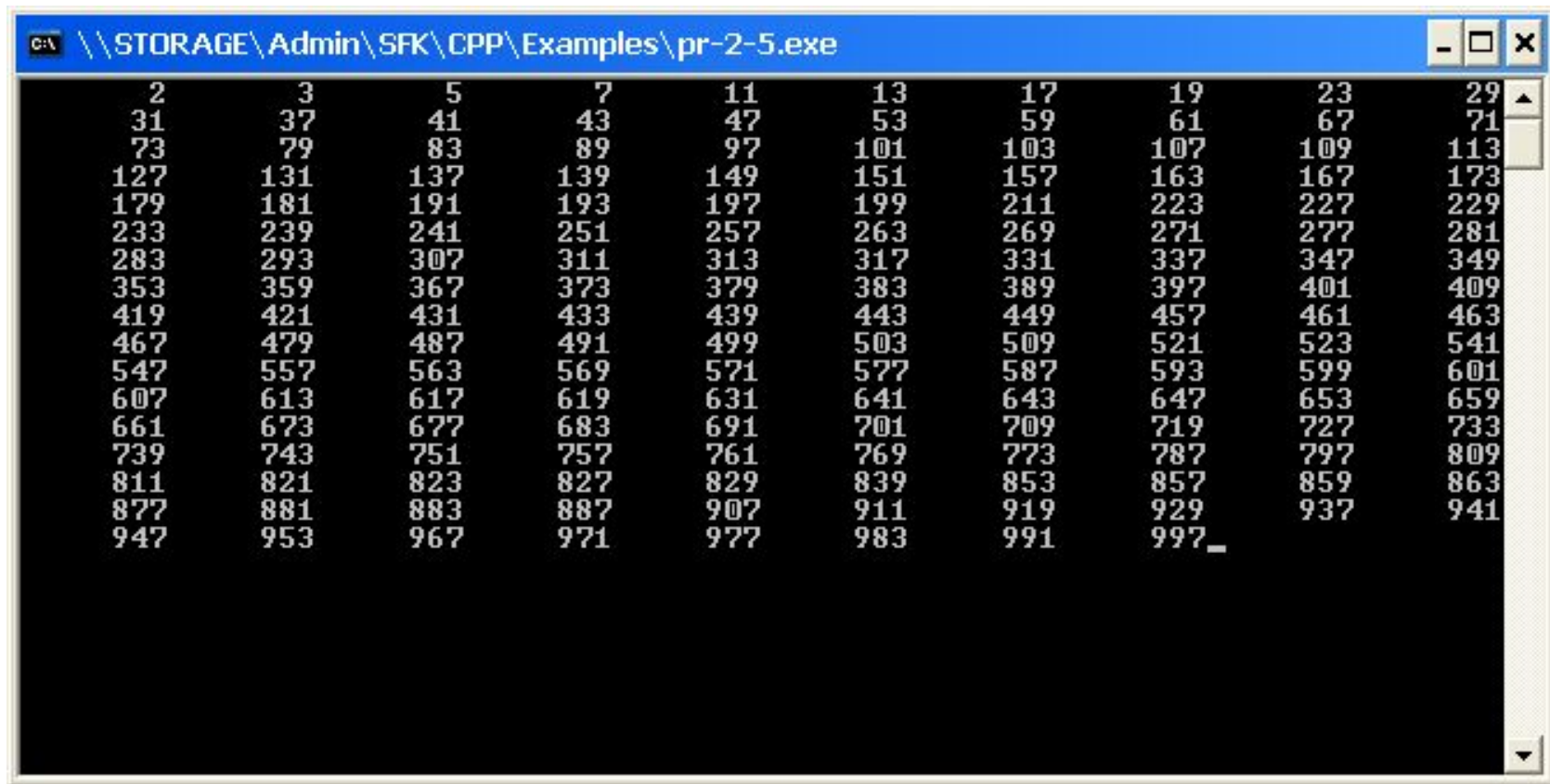


Операторы break и continue

`break` – прерывает самый внутренний цикл, в котором он встретился

`continue` – прерывает только одну итерацию

Поиск простых чисел



The screenshot shows a Windows command prompt window with the title bar "C:\ \\STORAGE\Admin\SFK\CPP\Examples\pr-2-5.exe". The window contains a list of prime numbers arranged in 10 columns. The numbers are: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997.

```
    }  
    if (b) printf("%8d",n) ;  
}  
}
```

Оператор switch

```
switch (n)
{
    case 2: printf("Плохо\n");
            printf("Вы отчислены\n");
            break;
    case 3: printf("Удовлетворительно\n");
            break;
    case 4: printf("Хорошо\n");
            break;
    case 5: printf("Отлично\n");
            break;
    default: printf("Недопустимая оценка\n");
}
```