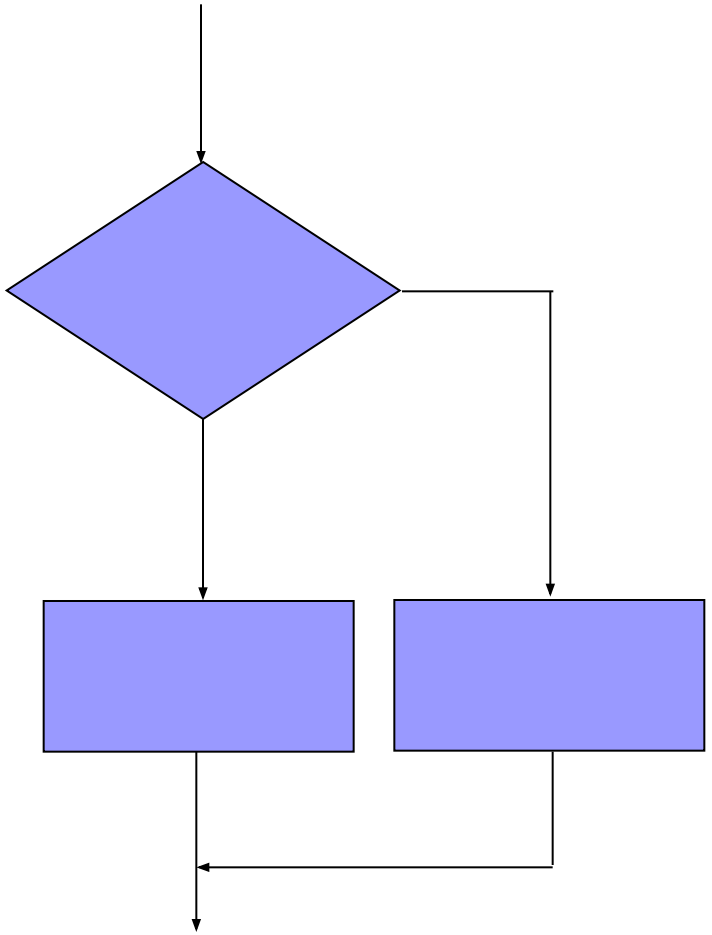
# Язык С

**Лекция №2**

Алгоритмические структуры

# Условный оператор

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

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



# Логические условия

- В языке С – целочисленная переменная (0 – ложь, не 0 – истина)
- В языке С++ 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 !

! (A && B) ≡ (!A || !B)

! (A || B) ≡ (!A && !B)

законы де Моргана

# Виды оператора **if**

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

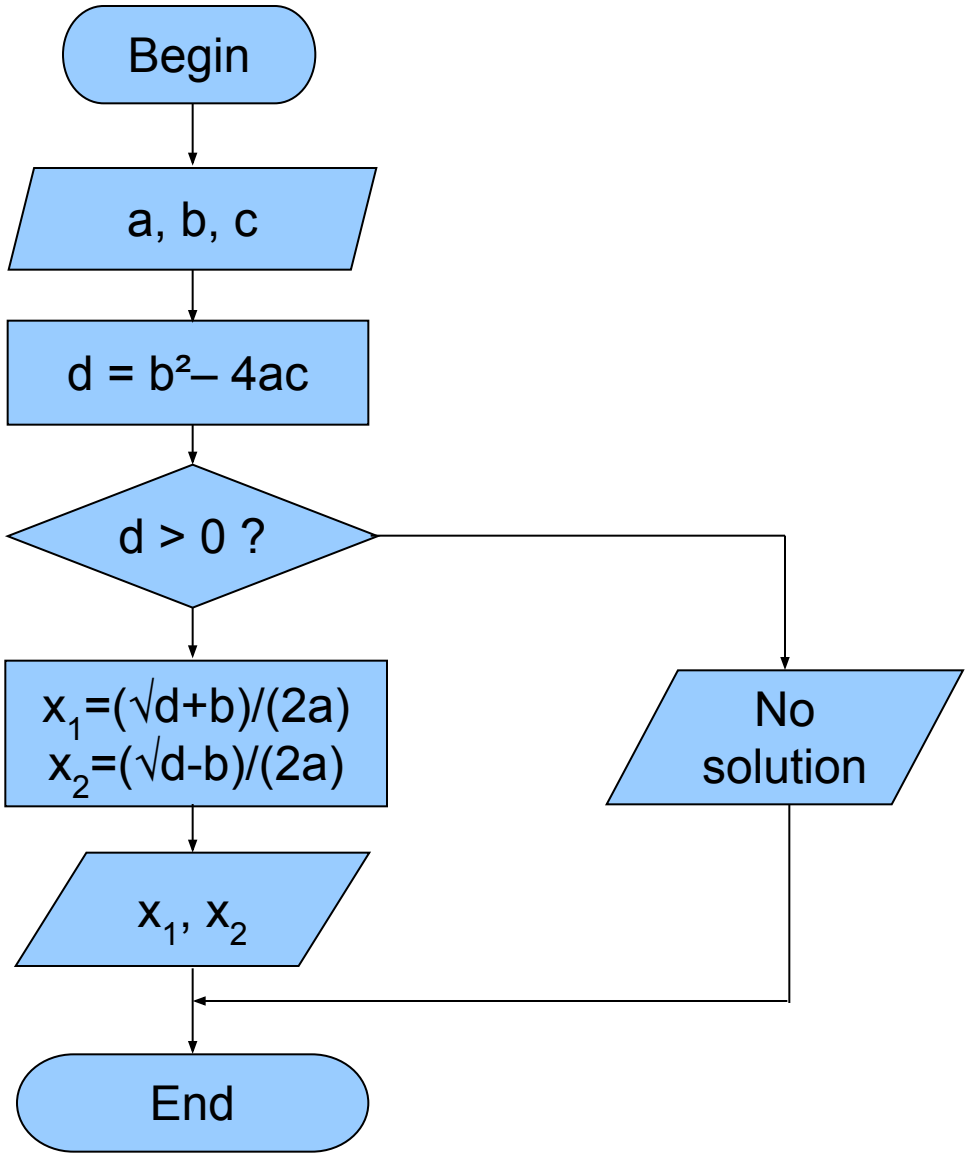
```
if (x>0) оператор;
else оператор;
```

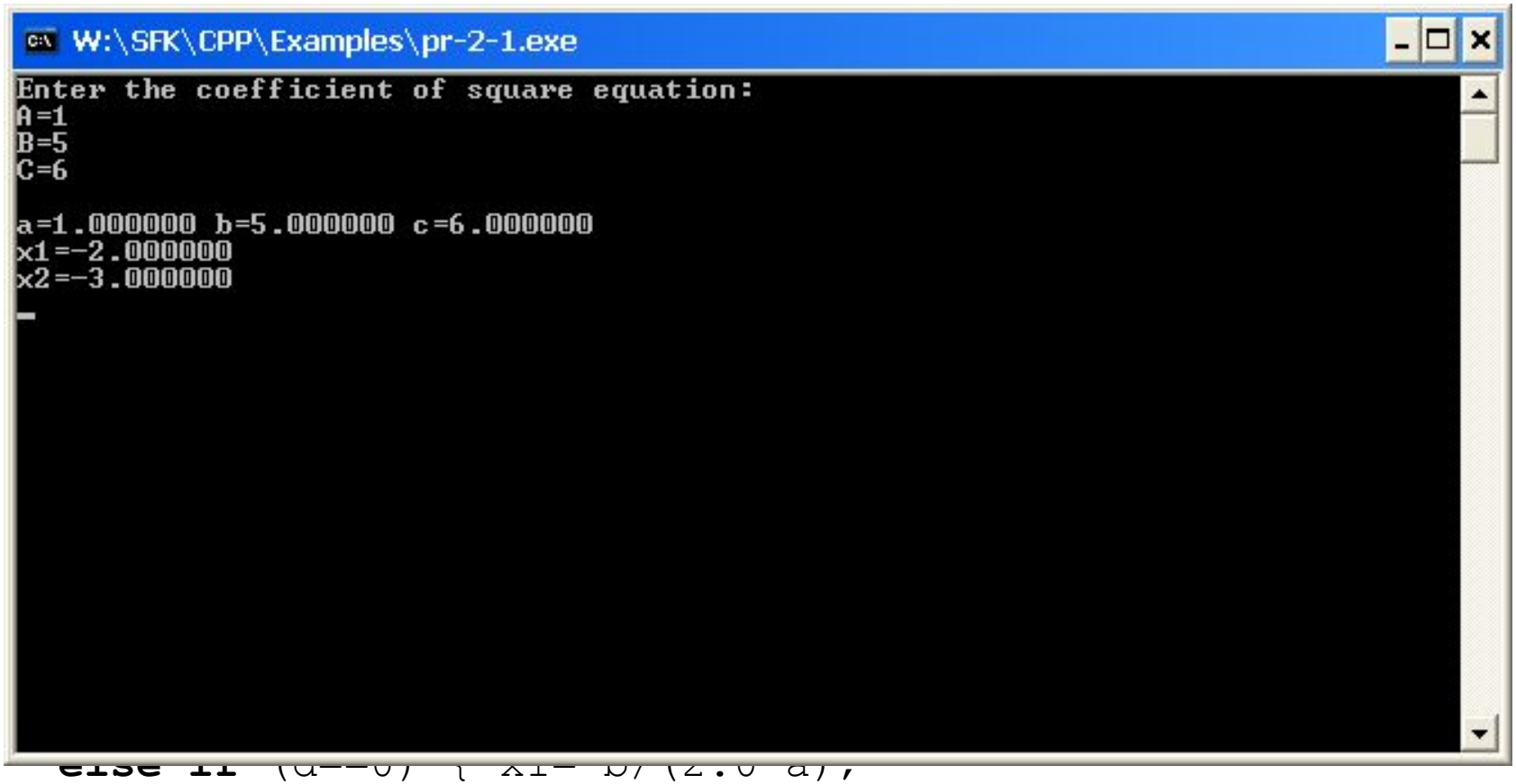
```
if (x == 0)
     { оператор;
       оператор;
     }
else
     {  оператор;
        оператор;
     }
```

# Вложенные операторы **if**

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

# Пример: решение квадратного уравнения



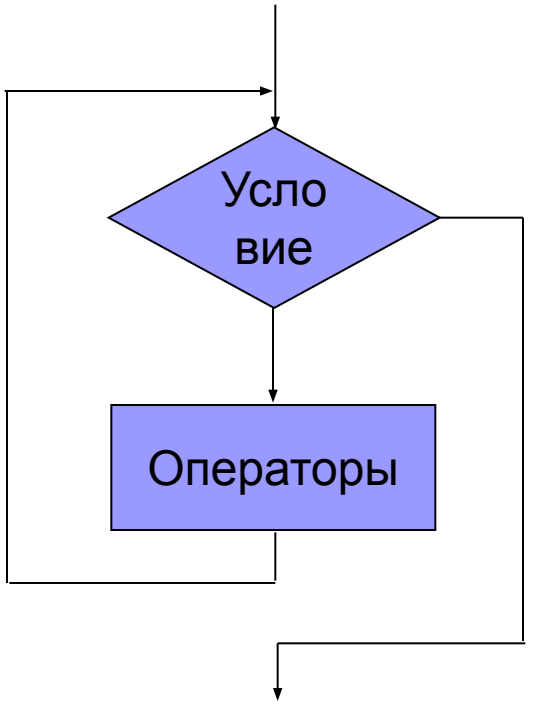



```
    else if (d==0) { x1=-b/(2.0*a);
            printf("x=%f\n",x1);
          }
      else       { d=sqrt(d); a=2.0*a;
           x1=(-b+d)/a; x2=(-b-d)/a;
                 printf("x1=%f\nx2=%f\n",x1,x2);
           }
}
```

# Цикл с предусловием

```
while (условие) оператор;
```

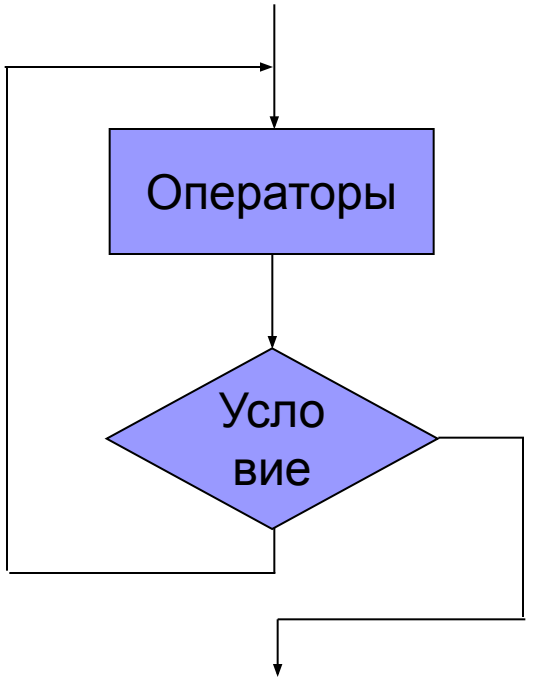
```
while (условие)
{
   оператор;
   оператор;
   оператор;
}
```



# Цикл с постусловием

```
do оператор while (условие);
```

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



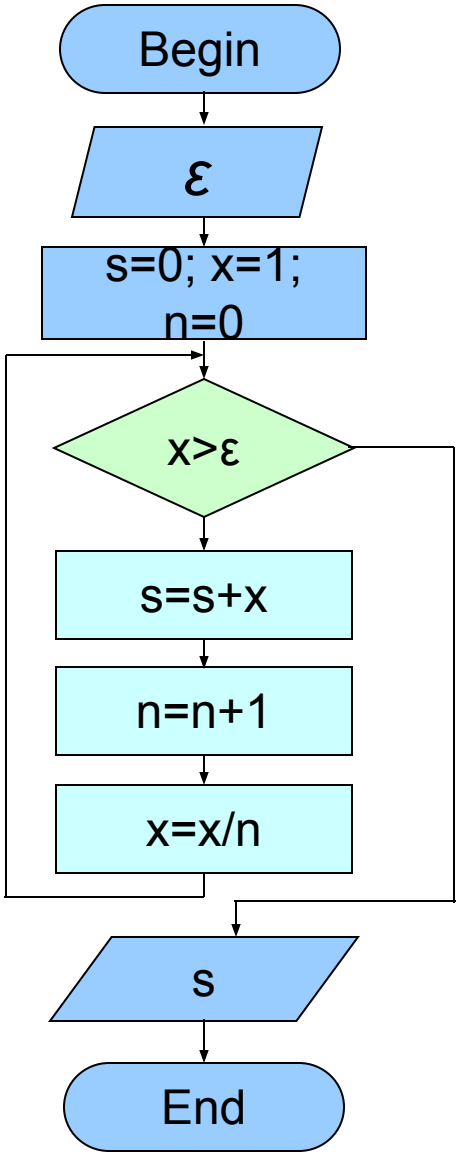
# Вычисление числа

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

s – накапливаемая сумма

x – член ряда

ε – требуемая точность



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

```
#include <stdio.h>
main () // Программа вычисления числа e
{
   double eps;    // Точность вычисления
   double x = 1.0;    // Член ряда
   double s = 0;      // Накапливаемая сумма
```

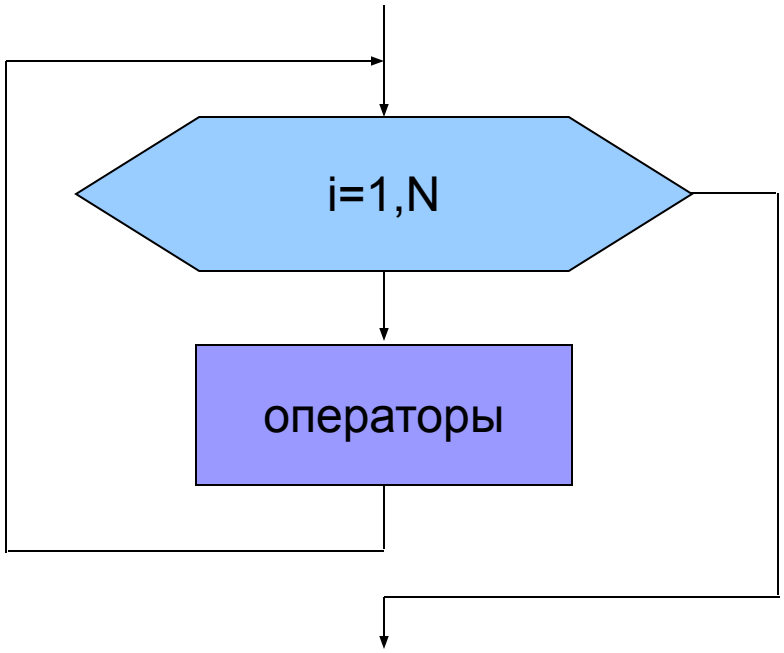
```
\\STORAGE\Admin\SFK\CPP\Examples\pr-2-2.exe
Enter the accuracy: 1E-9
eps=  1.000000E-009
 n= 0 x= 1.000000000000 s=  0.000000000000
 n= 1 x= 1.000000000000 s=  1.000000000000
 n= 2 x= 0.500000000000 s=  2.000000000000
 n= 3 x= 0.166666666667 s=  2.500000000000
 n= 4 x= 0.041666666667 s=  2.666666666667
 n= 5 x= 0.008333333333 s=  2.708333333333
 n= 6 x= 0.001388888889 s=  2.716666666667
 n= 7 x= 0.000198412698 s=  2.718055555556
 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
```

```
}
```

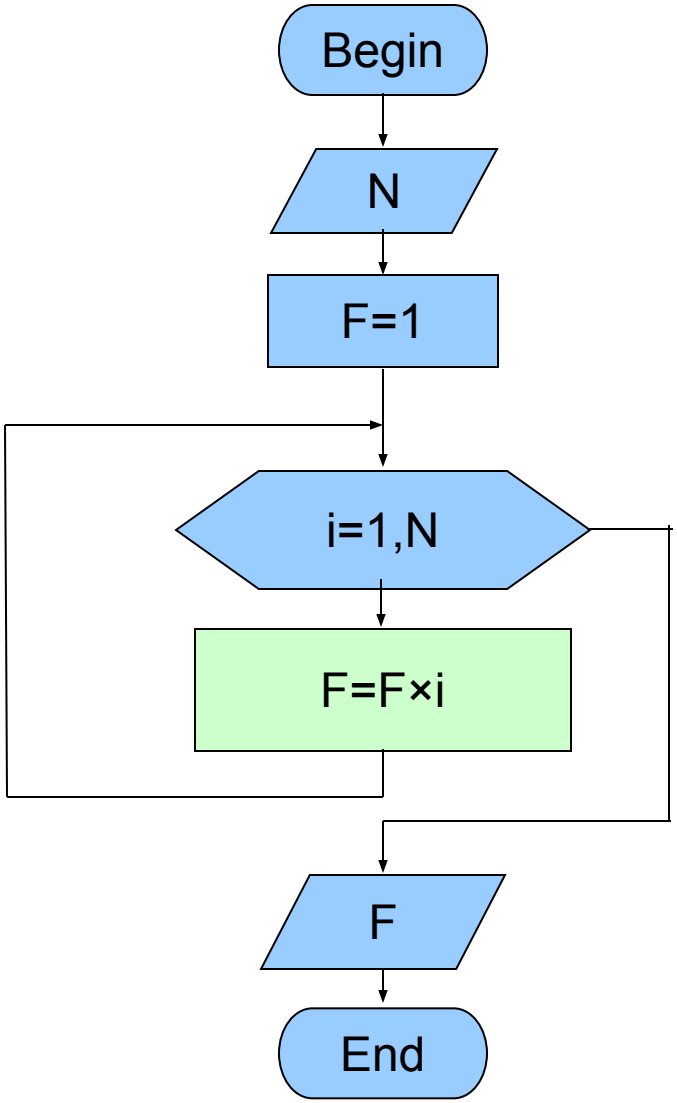
# Оператор цикла for

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

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



# Вычисление факториала $n! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot n$



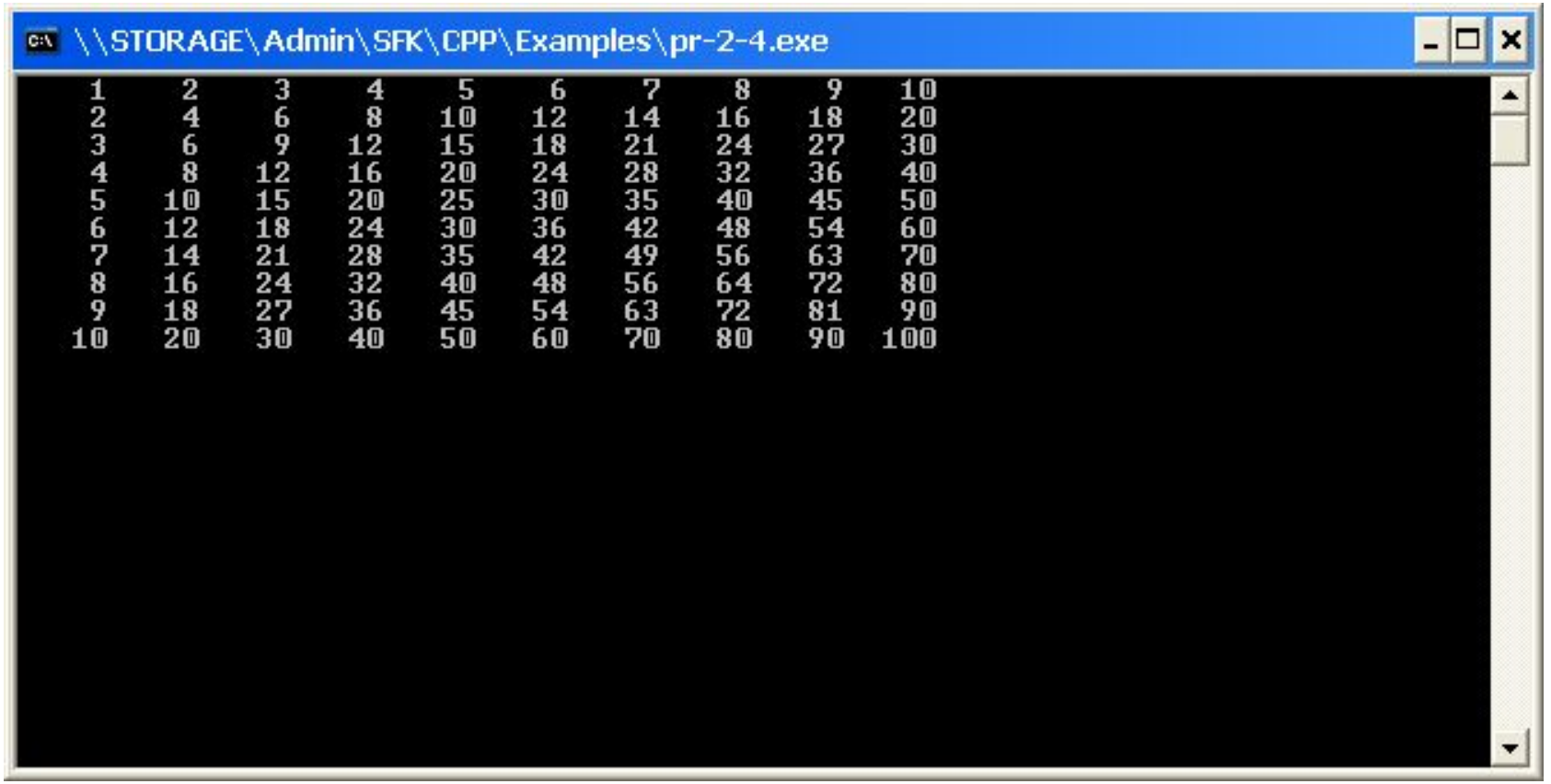
# Вычисление факториала

```
#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

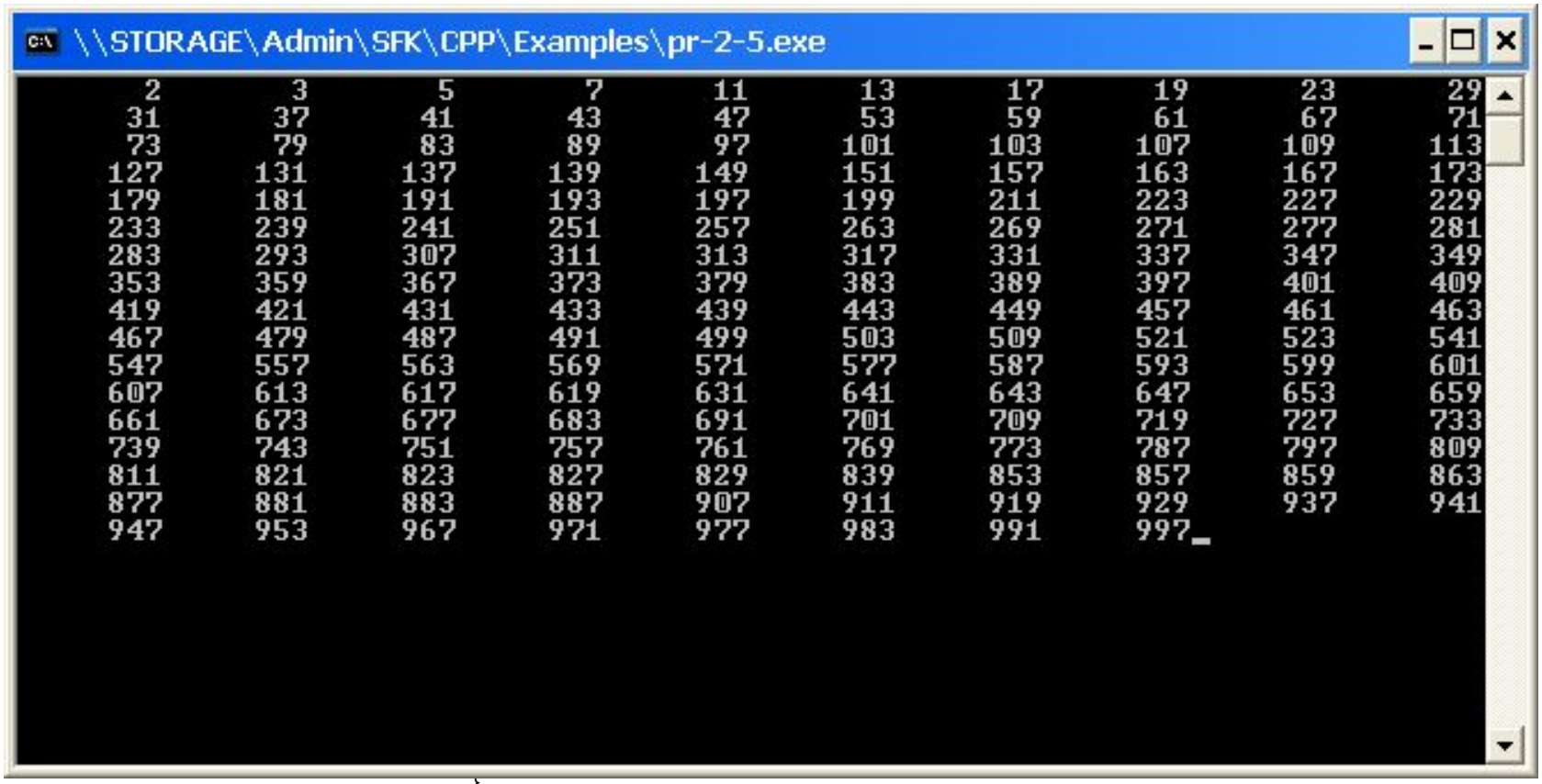


# Операторы break и continue

break – прерывает самый внутренний цикл, в котором он встретился

continue – прерывает только одну итерацию

# Поиск простых чисел




```
                    }
     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");
}
```