

Работа на ЭВМ и программирование (группа 114)

Занятие 2 (часть 2)

Контактная информация

- Шундеев Александр Сергеевич
- alex.shundeev@gmail.com
- <http://group112.github.io/sem1.html>

Электронная почта

- Тема письма

- 114 Фамилия Имя Отчество
- 114 Фамилия Имя

- Пример

- 114 Иванов Иван Иванович
- 114 Иванов Иван

Простые типы данных

Функция printf

```
#include <stdio.h>

int main(void)
{
    int x = 1;
    double y = 1.5;

    printf("x=%d x=%5d y=%f y=%5.1f y=%5.2f\n", x, x, y, y, y);

    return 0;
}
```

```
student@group112:~$ ./prog
```

```
x=1 x=      1 y=1.500000 y=  1.5 y= 1.50
```

```
student@group112:~$
```

Функция printf

```
printf("x=%d x=%5d y=%f y=%5.1f y=%5.2f\n", x, x, y, y, y);
```

форматная строка (строка формата печати)

Функция printf

```
printf("x=%d x=%5d y=%f y=%5.1f y=%5.2f\n", x, x, y, y, y);
```

специальные последовательности символов

Функция printf

```
printf("x=%d x=%5d y=%f y=%5.1f y=%5.2f\n", x, x, y, y, y);
```

Спецификаторы формата печати:

%d для int

%f для double

Функция printf

```
printf("x=%d x=%5d y=%f y=%5.1f y=%5.2f\n", x, x, y, y, y);
```

минимальная ширина поля печати

Функция printf

```
printf("x=%d x=%5d y=%f y=%5.1f y=%5.2f\n", x, x, y, y, y);
```

точность (количество цифр после запятой)

Функция scanf

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

```
int main(void)
```

```
{
```

```
    int x, s;
```

```
    double y;
```

```
    printf("Input x, y: ");
```

```
    s = scanf("%d%lf", &x, &y);
```

```
    printf("s=%d x=%d y=%f\n", s, x, y);
```

```
    return 0;
```

```
}
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

Файл Правка Вид Терминал Вкладки Справка

student@group112:~\$./prog

Input x, y: 2 3.5

s=2 x=2 y=3.500000

student@group112:~\$./prog

Input x, y: 2 a

s=1 x=2 y=0.000000

student@group112:~\$./prog

Input x, y: a

s=0 x=266588320 y=0.000000

student@group112:~\$ █

Функция scanf

```
s = scanf("%d%lf", &x, &y);
```

форматная строка (строка формата ввода)

Функция scanf

```
s = scanf("%d%lf", &x, &y);
```

адреса переменных

Функция scanf

```
s = scanf("%d%lf", &x, &y);
```

Спецификаторы формата ввода:

%d для int

%lf для double

Функция scanf

```
s = scanf("%d%lf", &x, &y);
```

Функция `scanf` возвращает количество успешно обработанных спецификаторов ввода или константу EOF (-1)

Функция scanf

```
s = scanf("%d%lf", &x, &y);
```

s = 0, 1, 2, EOF (-1)

Файл Правка Вид Терминал Вкладки Справка

student@group112:~\$./prog

Input x, y: 2 3.5

s=2 x=2 y=3.500000

student@group112:~\$./prog

Input x, y: 2 a

s=1 x=2 y=~~0.000000~~**student@group112:~\$./prog**

Input x, y: a

s=0 x=~~266588320~~ y=~~0.000000~~**student@group112:~\$** █

Первая программа

Постановка задачи

Задача

Дано:

a - начальный член арифметической прогрессии;

d - разность арифметической прогрессии;

$S1(n)$ - сумма (по определению) n первых членов прогрессии;

$S2(n)$ - сумма (по формуле) n первых членов прогрессии.

Найти:

наименьшее n : $|S1(n) - S2(n)| > 10^{-7}$.

Файл Правка Вид Терминал Вкладки Справка

student@group112:~\$ mkdir first**student@group112:~\$** cd first**student@group112:~/first\$** pwd

/home/student/first

student@group112:~/first\$ gcc -Wall -Wextra -Wfloat-equal -Werror -pedantic -std=c99 first.c -o prog**student@group112:~/first\$** ls

first.c prog

student@group112:~/first\$./prog

Input a: 1.5

Input d: 0.001

Файл Правка Вид Терминал Вкладки Справка

```
n: 30828 s1: 521409.378000100 s2: 521409.378000000 r: 0.000000099535
n: 30829 s1: 521441.706000100 s2: 521441.706000000 r: 0.000000099477
n: 30830 s1: 521474.035000100 s2: 521474.035000000 r: 0.000000099535
n: 30831 s1: 521506.365000100 s2: 521506.365000000 r: 0.000000099593
n: 30832 s1: 521538.696000100 s2: 521538.696000000 r: 0.000000099535
n: 30833 s1: 521571.028000100 s2: 521571.028000000 r: 0.000000099593
n: 30834 s1: 521603.361000100 s2: 521603.361000000 r: 0.000000099652
n: 30835 s1: 521635.695000100 s2: 521635.695000000 r: 0.000000099593
n: 30836 s1: 521668.030000100 s2: 521668.030000000 r: 0.000000099652
n: 30837 s1: 521700.366000100 s2: 521700.366000000 r: 0.000000099710
n: 30838 s1: 521732.703000100 s2: 521732.703000000 r: 0.000000099652
n: 30839 s1: 521765.041000100 s2: 521765.041000000 r: 0.000000099652
n: 30840 s1: 521797.380000100 s2: 521797.380000000 r: 0.000000099710
n: 30841 s1: 521829.720000100 s2: 521829.720000000 r: 0.000000099710
n: 30842 s1: 521862.061000100 s2: 521862.061000000 r: 0.000000099710
n: 30843 s1: 521894.403000100 s2: 521894.403000000 r: 0.000000099768
n: 30844 s1: 521926.746000100 s2: 521926.746000000 r: 0.000000099710
n: 30845 s1: 521959.090000100 s2: 521959.090000000 r: 0.000000099768
n: 30846 s1: 521991.435000100 s2: 521991.435000000 r: 0.000000099826
n: 30847 s1: 522023.781000100 s2: 522023.781000000 r: 0.000000099768
n: 30848 s1: 522056.128000100 s2: 522056.128000000 r: 0.000000099826
n: 30849 s1: 522088.476000100 s2: 522088.476000000 r: 0.000000099884
n: 30850 s1: 522120.825000100 s2: 522120.825000000 r: 0.000000099768
n: 30851 s1: 522153.175000100 s2: 522153.175000000 r: 0.000000099826
n: 30852 s1: 522185.526000100 s2: 522185.526000000 r: 0.000000099884
n: 30853 s1: 522217.878000100 s2: 522217.878000000 r: 0.000000099826
n: 30854 s1: 522250.231000100 s2: 522250.231000000 r: 0.000000099884
n: 30855 s1: 522282.585000100 s2: 522282.585000000 r: 0.000000099943
n: 30856 s1: 522314.940000100 s2: 522314.940000000 r: 0.000000099884
n: 30857 s1: 522347.296000100 s2: 522347.296000000 r: 0.000000099943
```

Find n: 30858

student@group112:~/first\$

Команда компиляции

```
gcc -Wall -Wextra -Wfloat-equal -Werror -pedantic -std=c99 first.c -o prog
```

Принудительное завершение программы

Если

программа «зациклилась»,

программа «не отвечает»,

надоело ждать завершения программы,

то одновременно нажимаем две клавиши

<Ctrl>-C

Необходимо прислать

- Файл с текстом программы (например, first.c)
- Скриншоты:
 - создание директории
 - печать полного пути текущей директории
 - компиляция программы
 - содержимое директории
 - запуск программы и ввод параметров
 - результат выполнения программы

Первая программа

Файл с текстом программы

Структура файла first.txt

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

```
#define EPS 1e-7
```

```
#define MAX 100000
```

```
double sum1(double a, double d, int n);
```

```
double sum2(double a, double d, int n);
```

```
int main(void)
```

```
{
```

```
    ...
```

```
}
```

```
double sum1(double a, double d, int n)
```

```
{
```

```
    ...
```

```
}
```

```
double sum2(double a, double d, int n)
```

```
{
```

```
    ...
```

```
}
```

Структура файла first.txt

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

```
#define EPS 1e-7
```

```
#define MAX 100000
```

```
double sum1(double a, double d, int n);
```

```
double sum2(double a, double d, int n);
```

```
int main(void)
```

```
{
```

```
    ...
```

```
}
```

```
double sum1(double a, double d, int n)
```

```
{
```

```
    ...
```

```
}
```

```
double sum2(double a, double d, int n)
```

```
{
```

```
    ...
```

```
}
```

Выполнение программы начинается с функции
main

Структура файла first.txt

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

```
#define EPS 1e-7
```

```
#define MAX 100000
```

```
double sum1(double a, double d, int n);
```

```
double sum2(double a, double d, int n);
```

```
int main(void)
```

```
{
```

```
    ...
```

```
}
```

```
double sum1(double a, double d, int n)
```

```
{
```

```
    ...
```

```
}
```

```
double sum2(double a, double d, int n)
```

```
{
```

```
    ...
```

```
}
```

Внутри функции **main** вызываются функции **sum1** и **sum2**

Структура файла first.txt

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

```
#define EPS 1e-7
```

```
#define MAX 100000
```

```
double sum1(double a, double d, int n);
```

```
double sum2(double a, double d, int n);
```

```
int main(void)
```

```
{
```

```
    ...
```

```
}
```

```
double sum1(double a, double d, int n)
```

```
{
```

```
    ...
```

```
}
```

```
double sum2(double a, double d, int n)
```

```
{
```

```
    ...
```

```
}
```

Перед использованием функции должны быть объявлены

Структура файла first.txt

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

```
#define EPS 1e-7  
#define MAX 100000
```

```
double sum1(double a, double d, int n);  
double sum2(double a, double d, int n);
```

```
int main(void)  
{  
    ...  
}
```

```
double sum1(double a, double d, int n)  
{  
    ...  
}
```

```
double sum2(double a, double d, int n)  
{  
    ...  
}
```

Директива препроцессора **#include** - вставка содержимого файла

Структура файла first.txt

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

```
#define EPS 1e-7
```

```
#define MAX 100000
```

```
double sum1(double a, double d, int n);
```

```
double sum2(double a, double d, int n);
```

```
int main(void)
```

```
{
```

```
    ...
```

```
}
```

```
double sum1(double a, double d, int n)
```

```
{
```

```
    ...
```

```
}
```

```
double sum2(double a, double d, int n)
```

```
{
```

```
    ...
```

```
}
```

Директива препроцессора **#define** - определение констант

Функция sum1 (вариант 1)

```
double sum1(double a, double d, int n);
```

```
...
```

```
double sum1(double a, double d, int n)
```

```
{
```

```
    double sum, cur;
```

```
    int i;
```

```
    sum = 0.;
```

```
    cur = a;
```

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

```
    {
```

```
        sum += cur;
```

```
        cur += d;
```

```
    }
```

```
    return sum;
```

```
}
```

Функция sum1 (вариант 1)

```
double sum1(double a, double d, int n);
```

Заголовок функции

```
...
```

```
double sum1(double a, double d, int n)
```

```
{
```

```
    double sum, cur;
```

```
    int i;
```

```
    sum = 0.;
```

```
    cur = a;
```

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

```
    {
```

```
        sum += cur;
```

```
        cur += d;
```

```
    }
```

```
    return sum;
```

```
}
```

Функция sum1 (вариант 1)

```
double sum1(double a, double d, int n);
```

```
...
```

```
double sum1(double a, double d, int n)
```

```
{  
    double sum, cur;  
    int i;  
  
    sum = 0.;  
    cur = a;  
  
    for(i = 1; i <= n; i++)  
    {  
        sum += cur;  
        cur += d;  
    }  
  
    return sum;  
}
```

Тело функции

Функция sum1 (вариант 1)

```
double sum1(double a, double d, int n);
```

Прототип функции

...

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    sum = 0.;
    cur = a;

    for(i = 1; i <= n; i++)
    {
        sum += cur;
        cur += d;
    }

    return sum;
}
```

Функция sum1 (вариант 1)

```
double sum1(double a, double d, int n);
```

```
...
```

```
double sum1(double a, double d, int n)
```

```
{
```

```
    double sum, cur;
```

```
    int i;
```

```
    sum = 0.;
```

```
    cur = a;
```

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

```
    {
```

```
        sum += cur;
```

```
        cur += d;
```

```
    }
```

```
    return sum;
```

```
}
```

Структура заголовка функции:

- имя функции

Функция sum1 (вариант 1)

```
double sum1(double a, double d, int n);
```

```
...
```

```
double sum1(double a, double d, int n)
```

```
{
```

```
    double sum, cur;
```

```
    int i;
```

```
    sum = 0.;
```

```
    cur = a;
```

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

```
    {
```

```
        sum += cur;
```

```
        cur += d;
```

```
    }
```

```
    return sum;
```

```
}
```

Структура заголовка функции:

- тип возвращаемого значения или void

Функция sum1 (вариант 1)

```
double sum1(double a, double d, int n);
```

```
...
```

```
double sum1(double a, double d, int n)
```

```
{
```

```
    double sum, cur;
```

```
    int i;
```

```
    sum = 0.;
```

```
    cur = a;
```

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

```
    {
```

```
        sum += cur;
```

```
        cur += d;
```

```
    }
```

```
    return sum;
```

```
}
```

Структура заголовка функции:

- список входных параметров или void

Функция sum1 (вариант 1)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    sum = 0.;
    cur = a;

    for(i = 1; i <= n; i++)
    {
        sum += cur;
        cur += d;
    }

    return sum;
}
```

Определение переменных

Функция sum1 (вариант 1)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    sum = 0.;
    cur = a;

    for(i = 1; i <= n; i++)
    {
        sum += cur;
        cur += d;
    }

    return sum;
}
```

Присваивание значений переменным

Функция sum1 (вариант 1)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    sum = 0.;
    cur = a;

    for(i = 1; i <= n; i++)
    {
        sum += cur;
        cur += d;
    }

    return sum;
}
```

Оператор цикла

`for(<init>;<test>;<incr>) <body>`

- init - начальное выражение
- test - проверочное выражение
- incr - выражение перехода
- body - тело цикла

Функция sum1 (вариант 1)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    sum = 0.;
    cur = a;

    for(i = 1; i <= n; i++)
    {
        sum += cur;
        cur += d;
    }

    return sum;
}
```

Оператор цикла

for(<init>;<test>;<incr>) <body>

- **init** - начальное выражение
- **test** - проверочное выражение
- **incr** - выражение перехода
- **body** - тело цикла

Функция sum1 (вариант 1)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    sum = 0.;
    cur = a;

    for(i = 1; i <= n; i++)
    {
        sum += cur;
        cur += d;
    }

    return sum;
}
```

Оператор цикла

for(<init>;<test>;<incr>) <body>

- init - начальное выражение
- test - проверочное выражение
- incr - выражение перехода
- body - тело цикла

Функция sum1 (вариант 1)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    sum = 0.;
    cur = a;

    for(i = 1; i <= n; i++)
    {
        sum += cur;
        cur += d;
    }

    return sum;
}
```

Оператор цикла

for(<init>;<test>;<incr>) <body>

- init - начальное выражение
- test - проверочное выражение
- incr - выражение перехода
- body - тело цикла

Функция sum1 (вариант 1)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    sum = 0.;
    cur = a;

    for(i = 1; i <= n; i++)
    {
        sum += cur;
        cur += d;
    }

    return sum;
}
```

Оператор цикла

for(<init>;<test>;<incr>) <body>

- init - начальное выражение
- test - проверочное выражение
- incr - выражение перехода
- body - тело цикла

Функция sum1 (вариант 1)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    sum = 0.;
    cur = a;

    for(i = 1; i <= n; i++)
    {
        sum += cur;
        cur += d;
    }

    return sum;
}
```

Оператор цикла

for(<init>;<test>;<incr>) <body>

- init - начальное выражение
- test - проверочное выражение
- incr - выражение перехода
- body - тело цикла

Функция sum1 (вариант 1)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    sum = 0.;
    cur = a;

    for(i = 1; i <= n; i++)
    {
        sum += cur;
        cur += d;
    }

    return sum;
}
```

Оператор return - выход из функции

Функция sum1 (вариант 2)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    for(i = 1, sum = 0., cur = a; i <= n; i ++, cur += d)
        sum += cur;

    return sum;
}
```

Функция sum1 (вариант 2)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    for(i = 1, sum = 0., cur = a; i <= n; i ++, cur += d)
        sum += cur;

    return sum;
}
```

Определение переменных

Функция sum1 (вариант 2)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    for(i = 1, sum = 0., cur = a; i <= n; i ++, cur += d)
        sum += cur;

    return sum;
}
```

Оператор цикла

`for(<init>;<test>;<incr>) <body>`

- `init` - начальное выражение
- `test` - проверочное выражение
- `incr` - выражение перехода
- `body` - тело цикла

Функция sum1 (вариант 2)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    for(i = 1, sum = 0., cur = a; i <= n; i ++, cur += d)
        sum += cur;

    return sum;
}
```

Оператор цикла

for(<init>;<test>;<incr>) <body>

- **init** - начальное выражение
- **test** - проверочное выражение
- **incr** - выражение перехода
- **body** - тело цикла

Функция sum1 (вариант 2)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    for(i = 1, sum = 0., cur = a; i <= n; i ++, cur += d)
        sum += cur;

    return sum;
}
```

Оператор цикла

for(<init>;<test>;<incr>) <body>

- init - начальное выражение
- test - проверочное выражение
- incr - выражение перехода
- body - тело цикла

Функция sum1 (вариант 2)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    for(i = 1, sum = 0., cur = a; i <= n; i ++, cur += d)
        sum += cur;

    return sum;
}
```

Оператор цикла

for(<init>;<test>;<incr>) <body>

- init - начальное выражение
- test - проверочное выражение
- incr - выражение перехода
- body - тело цикла

Функция sum1 (вариант 2)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    for(i = 1, sum = 0., cur = a; i <= n; i ++, cur += d)
        sum += cur;

    return sum;
}
```

Оператор цикла

for(<init>;<test>;<incr>) <body>

- init - начальное выражение
- test - проверочное выражение
- incr - выражение перехода
- body - тело цикла

Функция sum1 (вариант 2)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    for(i = 1, sum = 0., cur = a; i <= n; i ++, cur += d)
        sum += cur;

    return sum;
}
```

Оператор цикла

for(<init>;<test>;<incr>) <body>

- init - начальное выражение
- test - проверочное выражение
- incr - выражение перехода
- body - тело цикла

Функция sum1 (вариант 2)

```
double sum1(double a, double d, int n)
{
    double sum, cur;
    int i;

    for(i = 1, sum = 0., cur = a; i <= n; i ++, cur += d)
        sum += cur;

    return sum;
}
```

Оператор **return** - выход из функции

Функция sum2

```
double sum2(double a, double d, int n)
{
    return (2 * a + (n - 1) * d) * n / 2;
}
```

Функция sum2

```
double sum2(double a, double d, int n)
{
    return (2 * a + (n - 1) * d) * n / 2;
}
```

Оператор `return` - выход из функции

Функция main (начало)

```
int main(void)
{
    double a, d, s1, s2, r;
    int n;

    printf("Input a: ");
    if(scanf("%lf", &a) != 1)
    {
        printf("Can't read a ...\n");
        return -1;
    }

    printf("Input d: ");
    if(scanf("%lf", &d) != 1)
    {
        printf("Can't read d ...\n");
        return -1;
    }
}
```

Функция main (начало)

```
int main(void)
{
    double a, d, s1, s2, r;
    int n;

    printf("Input a: ");
    if(scanf("%lf", &a) != 1)
    {
        printf("Can't read a ...\n");
        return -1;
    }

    printf("Input d: ");
    if(scanf("%lf", &d) != 1)
    {
        printf("Can't read d ...\n");
        return -1;
    }
}
```

Определение переменных

Функция main (начало)

```
int main(void)
{
    double a, d, s1, s2, r;
    int n;

    printf("Input a: ");
    if(scanf("%lf", &a) != 1)
    {
        printf("Can't read a ...\n");
        return -1;
    }

    printf("Input d: ");
    if(scanf("%lf", &d) != 1)
    {
        printf("Can't read d ...\n");
        return -1;
    }
}
```

Печатается приглашение ввести входной параметр

Функция main (начало)

```
int main(void)
{
    double a, d, s1, s2, r;
    int n;

    printf("Input a: ");
    if(scanf("%lf", &a) != 1)
    {
        printf("Can't read a ...\n");
        return -1;
    }

    printf("Input d: ");
    if(scanf("%lf", &d) != 1)
    {
        printf("Can't read d ...\n");
        return -1;
    }
}
```

Условный оператор

if(<test>) <oper>

- test - проверочное выражение
- oper - оператор

Функция main (начало)

```
int main(void)
{
    double a, d, s1, s2, r;
    int n;

    printf("Input a: ");
    if (scanf("%lf", &a) != 1)
    {
        printf("Can't read a ...\n");
        return -1;
    }

    printf("Input d: ");
    if (scanf("%lf", &d) != 1)
    {
        printf("Can't read d ...\n");
        return -1;
    }
}
```

Условный оператор

if(<test>) <oper>

- test - проверочное выражение
- oper - оператор

Функция main (начало)

```
int main(void)
{
    double a, d, s1, s2, r;
    int n;

    printf("Input a: ");
    if(scanf("%lf", &a) != 1)
    {
        printf("Can't read a ...\n");
        return -1;
    }

    printf("Input d: ");
    if(scanf("%lf", &d) != 1)
    {
        printf("Can't read d ...\n");
        return -1;
    }
}
```

Условный оператор

if(<test>) <oper>

- test - проверочное выражение = истина
- oper - оператор

Функция main (начало)

```
int main(void)
{
    double a, d, s1, s2, r;
    int n;

    printf("Input a: ");
    if(scanf("%lf", &a) != 1)
    {
        printf("Can't read a ...\n");
        return -1;
    }

    printf("Input d: ");
    if(scanf("%lf", &d) != 1)
    {
        printf("Can't read d ...\n");
        return -1;
    }
}
```

Печатается приглашение ввести входной параметр

Функция main (начало)

```
int main(void)
{
    double a, d, s1, s2, r;
    int n;

    printf("Input a: ");
    if(scanf("%lf", &a) != 1)
    {
        printf("Can't read a ...\n");
        return -1;
    }

    printf("Input d: ");
    if(scanf("%lf", &d) != 1)
    {
        printf("Can't read d ...\n");
        return -1;
    }
}
```

Условный оператор

if(<test>) <oper>

- test - проверочное выражение
- oper - оператор

Функция main (начало)

```
int main(void)
{
    double a, d, s1, s2, r;
    int n;

    printf("Input a: ");
    if(scanf("%lf", &a) != 1)
    {
        printf("Can't read a ...\n");
        return -1;
    }

    printf("Input d: ");
    if(scanf("%lf", &d) != 1)
    {
        printf("Can't read d ...\n");
        return -1;
    }
}
```

Условный оператор

if(<test>) <oper>

- test - проверочное выражение
- oper - оператор

Функция main (начало)

```
int main(void)
{
    double a, d, s1, s2, r;
    int n;

    printf("Input a: ");
    if(scanf("%lf", &a) != 1)
    {
        printf("Can't read a ...\n");
        return -1;
    }

    printf("Input d: ");
    if(scanf("%lf", &d) != 1)
    {
        printf("Can't read d ...\n");
        return -1;
    }
}
```

Условный оператор

if(<test>) <oper>

- test - проверочное выражение = истина
- oper - оператор

Функция main (продолжение)

```
for(n = 1; n <= MAX; n++)
{
    s1 = sum1(a, d, n);
    s2 = sum2(a, d, n);
    r = s1 - s2;
    r = r > 0 ? r : -r;

    if(r > EPS)
    {
        printf("Find n: %d\n", n);
        break;
    }
    else
        printf("n: %5d s1: %20.9lf s2: %20.9lf r: %.12lf\n", n, s1, s2, r);
}

return 0;
}
```

Функция main (продолжение)

```
for(n = 1; n <= MAX; n++)
{
    s1 = sum1(a, d, n);
    s2 = sum2(a, d, n);
    r = s1 - s2;
    r = r > 0 ? r : -r;

    if(r > EPS)
    {
        printf("Find n: %d\n", n);
        break;
    }
    else
        printf("n: %5d s1: %20.9lf s2: %20.9lf r: %.12lf\n", n, s1, s2, r);
}

return 0;
}
```

Функция main (продолжение)

```
for(n = 1; n <= MAX; n++)
{
    s1 = sum1(a, d, n);
    s2 = sum2(a, d, n);
    r = s1 - s2;
    r = r > 0 ? r : -r;

    if(r > EPS)
    {
        printf("Find n: %d\n", n);
        break;
    }
    else
        printf("n: %5d s1: %20.9lf s2: %20.9lf r: %.12lf\n", n, s1, s2, r);
}

return 0;
}
```

Функция main (продолжение)

```
for(n = 1; n <= MAX; n++)
{
    s1 = sum1(a, d, n);
    s2 = sum2(a, d, n);
    r = s1 - s2;
    r = r > 0 ? r : -r;

    if(r > EPS)
    {
        printf("Find n: %d\n", n);
        break;
    }
    else
        printf("n: %5d s1: %20.9lf s2: %20.9lf r: %.12lf\n", n, s1, s2, r);
}

return 0;
}
```

Функция main (продолжение)

```
for(n = 1; n <= MAX; n++)
{
    s1 = sum1(a, d, n);
    s2 = sum2(a, d, n);
    r = s1 - s2;
    r = r > 0 ? r : -r;

    if(r > EPS)
    {
        printf("Find n: %d\n", n);
        break;
    }
    else
        printf("n: %5d s1: %20.9lf s2: %20.9lf r: %.12lf\n", n, s1, s2, r);
}

return 0;
}
```

Функция main (продолжение)

```
for(n = 1; n <= MAX; n++)
{
    s1 = sum1(a, d, n);
    s2 = sum2(a, d, n);
    r = s1 - s2;
    r = r > 0 ? r : -r;

    if(r > EPS)
    {
        printf("Find n: %d\n", n);
        break;
    }
    else
        printf("n: %5d s1: %20.9lf s2: %20.9lf r: %.12lf\n", n, s1, s2, r);
}

return 0;
}
```