

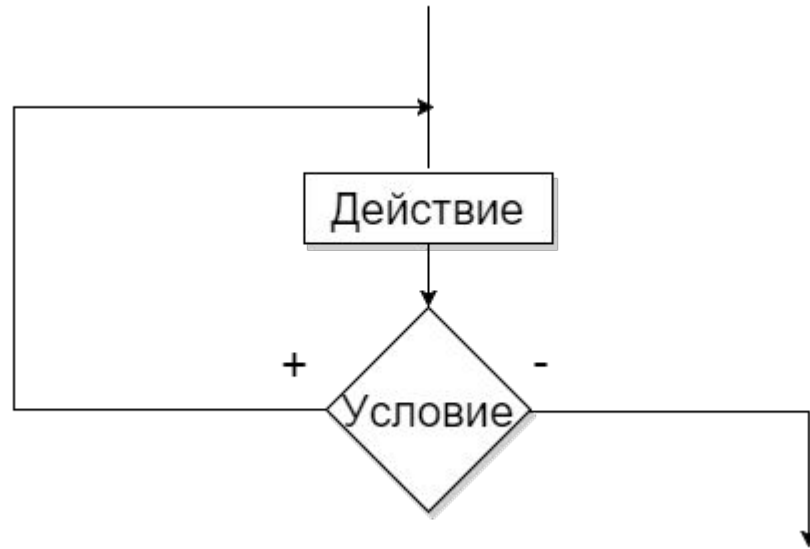
Основы программирования

Лабораторная работа №4

**Цикл DO
WHILE**

Власенко Олег Федосович

Цикл с постусловием do while

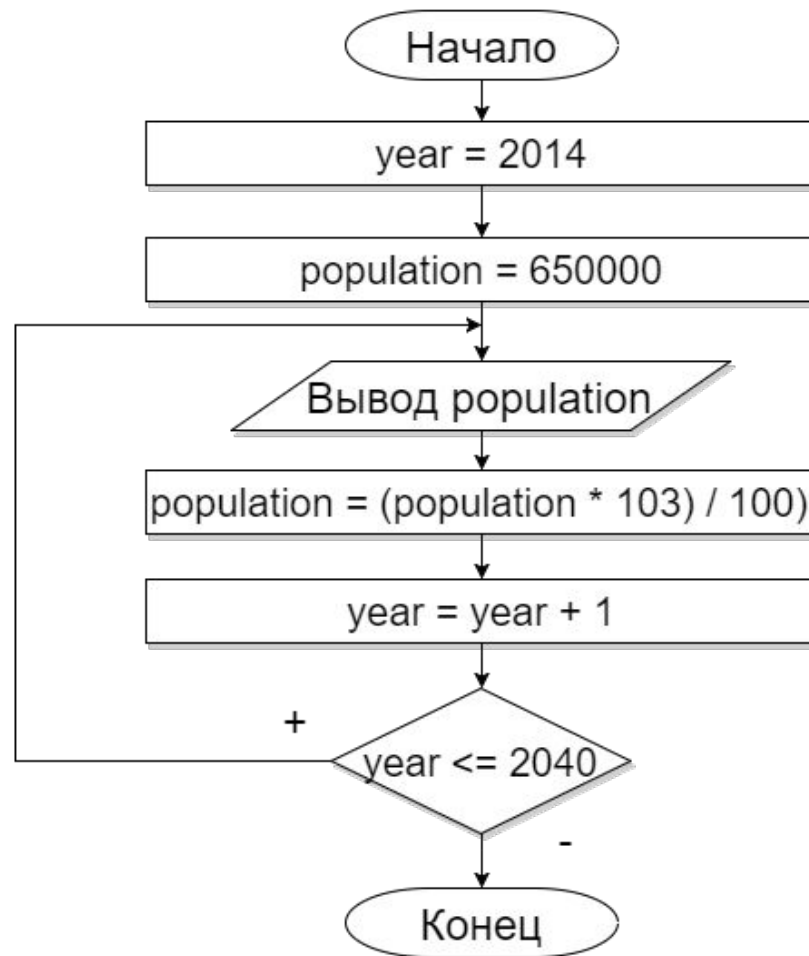


```
do {  
    Действие;  
} while (Условие);
```

Пример для цикла do while

Население города увеличивается на 3% каждый год. В 2014 году население города составляло 650 000 человек. Напишите программу, которая выведет на экран предсказываемую численность населения города в каждом году, вплоть до 2040.

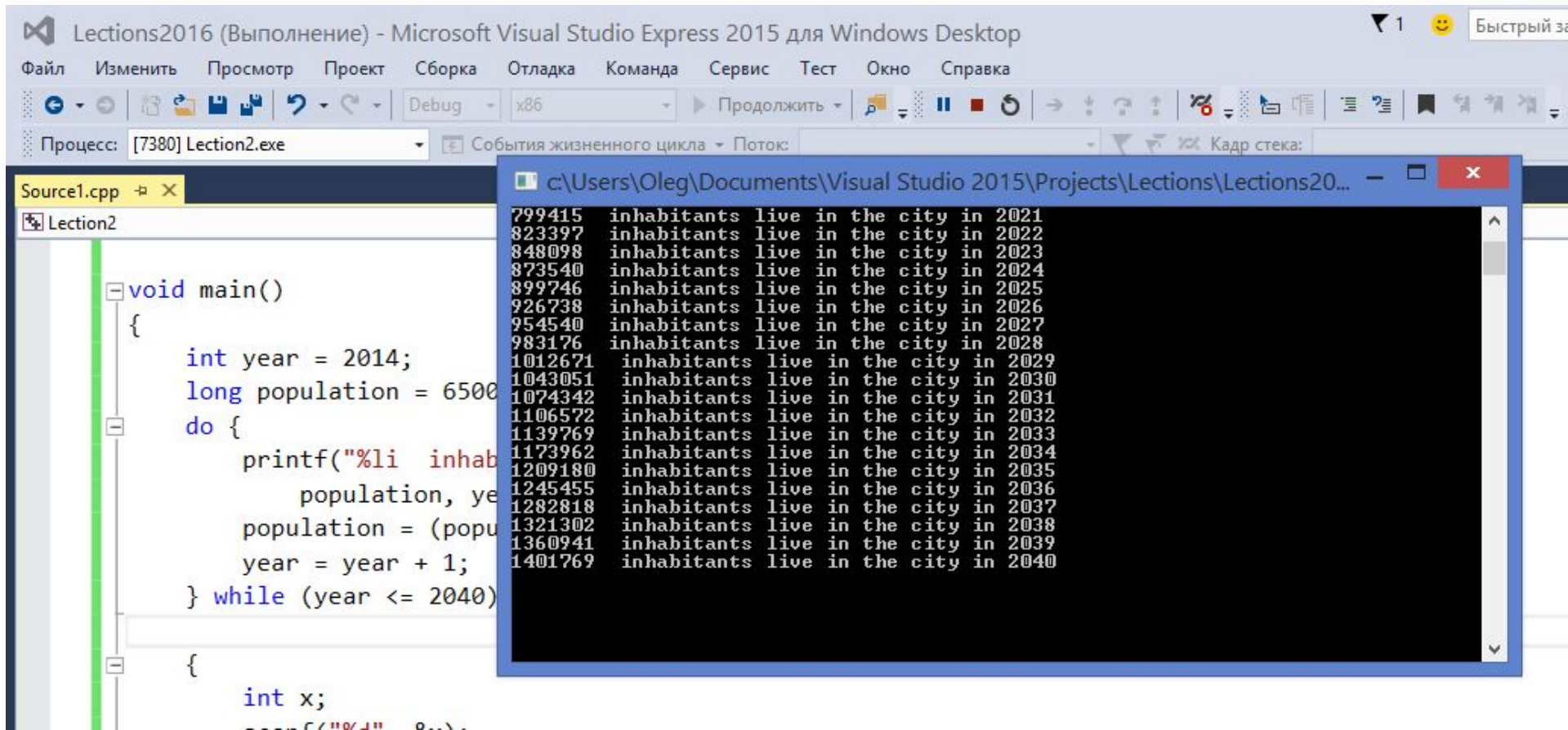
Блок-схема



Программа

```
void main() {  
    int year = 2014;  
    long population = 650000;  
    do {  
        printf("%li inhabitants live in the city in %i\n",  
            population, year);  
        population = (population * 103) / 100;  
        year = year + 1;  
    } while (year <= 2040);  
}
```

Программа в работе



The screenshot shows the Microsoft Visual Studio Express 2015 IDE. The main window displays the source code for 'Source1.cpp'. The code defines a 'main' function that initializes a year to 2014 and a population to 6500. It then enters a 'do-while' loop that prints the population for each year from 2021 to 2040, increments the year, and updates the population. The output window, titled 'c:\Users\Oleg\Documents\Visual Studio 2015\Projects\Lectons\Lectons20...', shows the results of the program's execution, displaying the population for each year from 2021 to 2040.

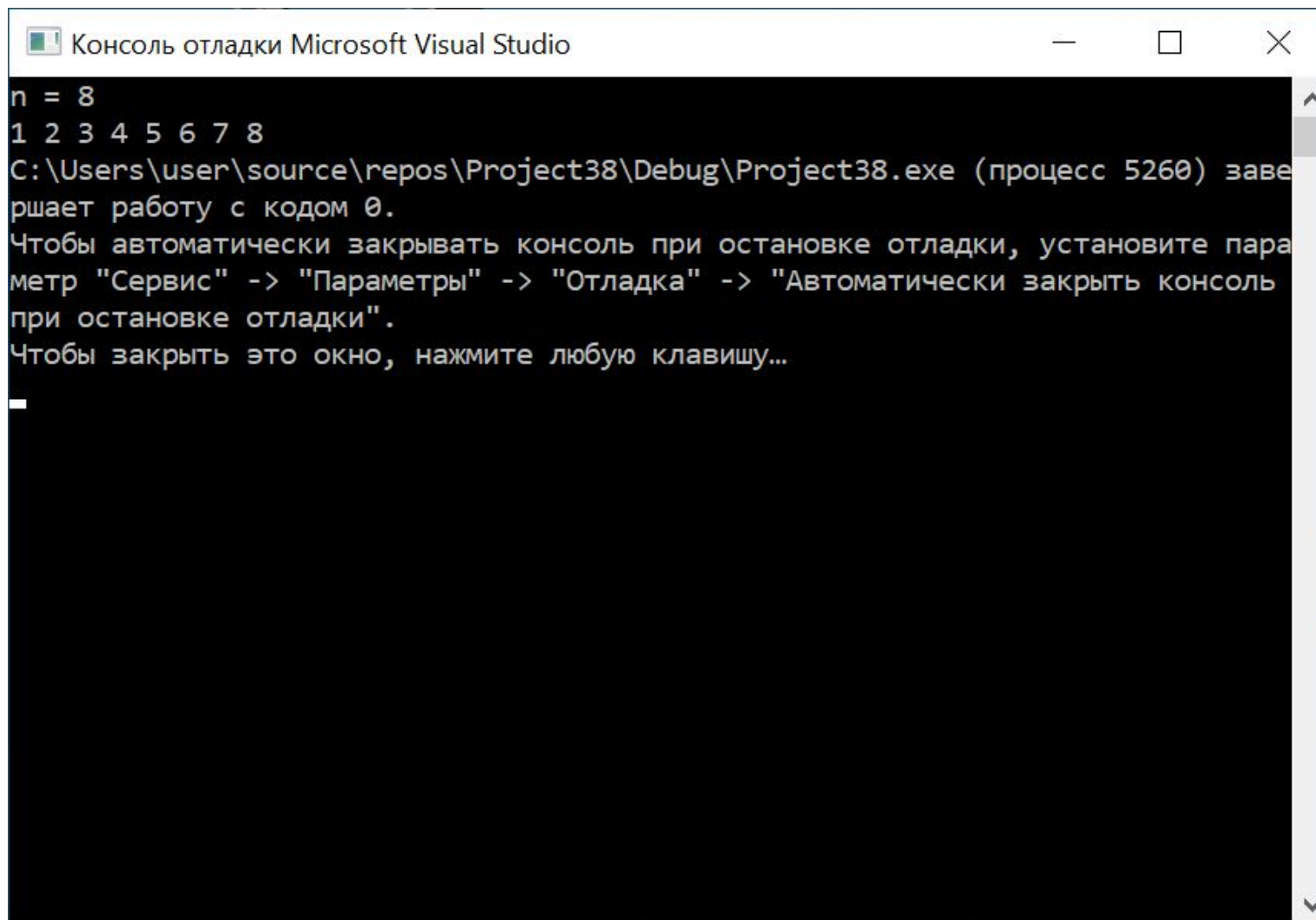
```
void main()
{
    int year = 2014;
    long population = 6500;
    do {
        printf("%li inhabitants live in the city in %d\n",
            population, year);
        population = (population * 1.02);
        year = year + 1;
    } while (year <= 2040);
}
```

```
799415 inhabitants live in the city in 2021
823397 inhabitants live in the city in 2022
848098 inhabitants live in the city in 2023
873540 inhabitants live in the city in 2024
899746 inhabitants live in the city in 2025
926738 inhabitants live in the city in 2026
954540 inhabitants live in the city in 2027
983176 inhabitants live in the city in 2028
1012671 inhabitants live in the city in 2029
1043051 inhabitants live in the city in 2030
1074342 inhabitants live in the city in 2031
1106572 inhabitants live in the city in 2032
1139769 inhabitants live in the city in 2033
1173962 inhabitants live in the city in 2034
1209180 inhabitants live in the city in 2035
1245455 inhabitants live in the city in 2036
1282818 inhabitants live in the city in 2037
1321302 inhabitants live in the city in 2038
1360941 inhabitants live in the city in 2039
1401769 inhabitants live in the city in 2040
```

Задача 1. Ряд натуральных чисел

Вводится N.

Нужно вывести натуральные числа от 1 до N (включительно).



```
Консоль отладки Microsoft Visual Studio

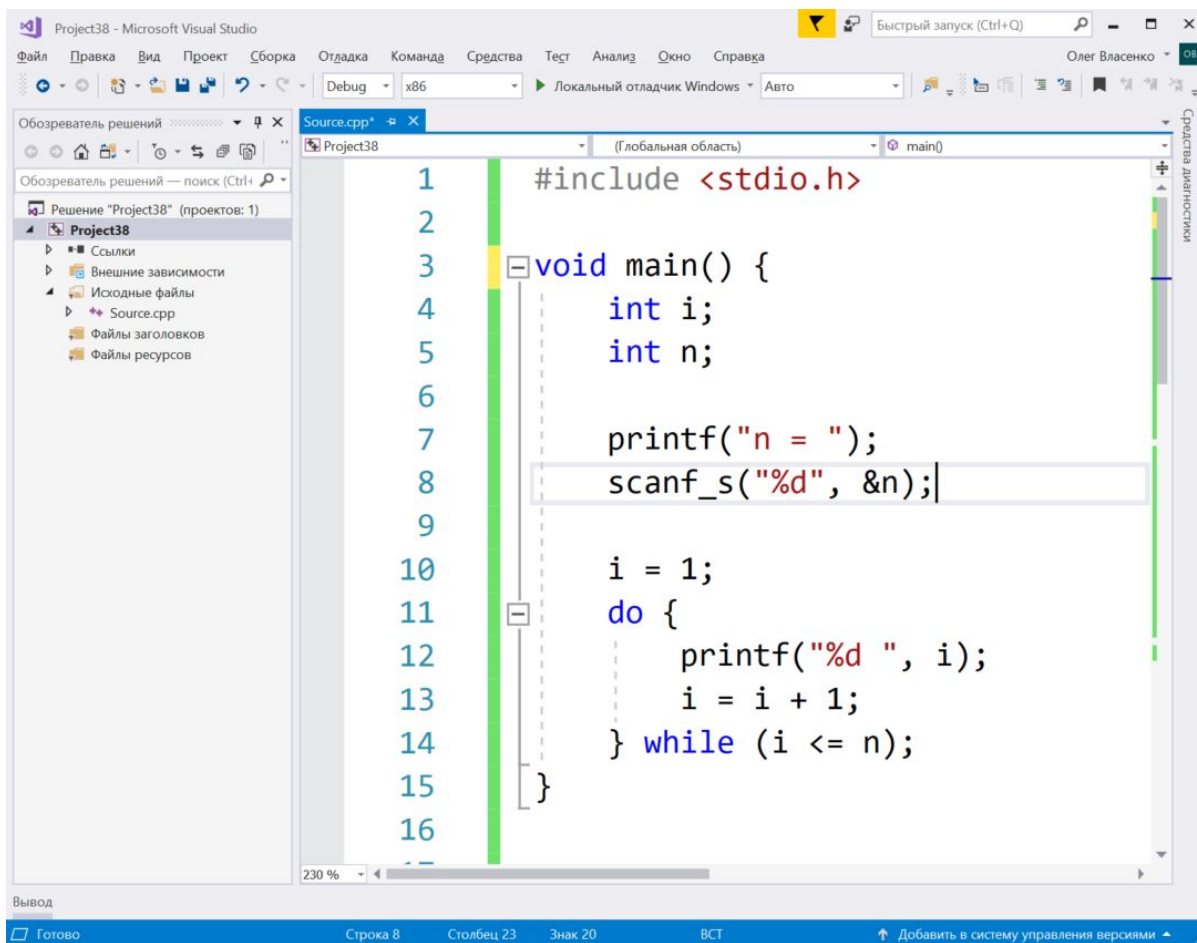
n = 8
1 2 3 4 5 6 7 8
C:\Users\user\source\repos\Project38\Debug\Project38.exe (процесс 5260) завершает работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, установите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Чтобы закрыть это окно, нажмите любую клавишу...
```

Задача 1. Ряд натуральных чисел

Вводится N.

Нужно вывести натуральные числа от 1 до N (включительно).

(Ниже – программа целиком)



The screenshot shows the Microsoft Visual Studio IDE with a C++ project named 'Project38'. The source file 'Source.cpp' is open, displaying the following code:

```
1  #include <stdio.h>
2
3  void main() {
4      int i;
5      int n;
6
7      printf("n = ");
8      scanf_s("%d", &n);
9
10     i = 1;
11     do {
12         printf("%d ", i);
13         i = i + 1;
14     } while (i <= n);
15 }
16
```

The code is written in C++ and uses the `scanf_s` function for input. It prints the numbers from 1 to `n` on a single line, separated by spaces. The IDE interface includes a menu bar, a toolbar, a Solution Explorer on the left, and a status bar at the bottom.

Задача 1. Ряд натуральных чисел – трассировка(0)

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

i = 1;
do {
    printf("%d ", i);
    i = i + 1;
} while (i <= n);
```

[illegible]

Задача 1. Ряд натуральных чисел – трассировка(1)

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

```
i = 1;
do {
    printf("%d ", i);
    i = i + 1;
} while (i <= n);
```

[illegible]

Задача 1. Ряд натуральных чисел – трассировка(2)

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

i = 1;
do {
    printf("%d ", i);
    i = i + 1;
} while (i <= n);
```

[illegible]

Задача 1. Ряд натуральных чисел – трассировка(3)

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

i = 1;
do {
    printf("%d ", i);
    i = i + 1;
} while (i <= n);
```

[illegible]

Задача 1. Ряд натуральных чисел – трассировка(4)

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

i = 1;
do {
    printf("%d ", i);
    i = i + 1;
} while (i <= n);
```

[illegible]

Задача 1. Ряд натуральных чисел – трассировка(5)

```
printf("n = ");  
scanf_s("%d", &n);  
  
i = 1;  
do {  
    printf("%d ", i);  
    i = i + 1;  
} while (i <= n);
```

i	n	i<=n	Вывод	Ввод
1	4		"n = "	4 <Enter>
			"1 "	

Задача 1. Ряд натуральных чисел – трассировка(6)

```
printf("n = ");  
scanf_s("%d", &n);  
  
i = 1;  
do {  
    printf("%d ", i);  
    i = i + 1;  
} while (i <= n);
```

i	n	i<=n	Вывод	Ввод
1	4		"n = "	4 <Enter>
2			"1 "	

Задача 1. Ряд натуральных чисел – трассировка(7)

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

i = 1;
do {
    printf("%d ", i);
    i = i + 1;
} while (i <= n);
```

i	n	i<=n	Вывод	Ввод
1	4		"n = "	4 <Enter>
2		2 <= 4 +	"1 "	

Задача 1. Ряд натуральных чисел – трассировка(8)

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

i = 1;
do {
    printf("%d ", i);
    i = i + 1;
} while (i <= n);
```

i	n	i<=n	Вывод	Ввод
1	4		"n = "	4 <Enter>
2		2 <= 4 +	"1 "	
			"2 "	

Задача 1. Ряд натуральных чисел – трассировка(9)

```
printf("n = ");  
scanf_s("%d", &n);  
  
i = 1;  
do {  
    printf("%d ", i);  
    i = i + 1;  
} while (i <= n);
```

i	n	i<=n	Вывод	Ввод
1	4		"n = "	4 <Enter>
2		2 <= 4 +	"1 "	
3			"2 "	

Задача 1. Ряд натуральных чисел – трассировка(10)

```
printf("n = ");  
scanf_s("%d", &n);  
  
i = 1;  
do {  
    printf("%d ", i);  
    i = i + 1;  
} while (i <= n);
```

i	n	i<=n	Вывод	Ввод
1	4		"n = "	4 <Enter>
2		2 <= 4 +	"1 "	
3		3 <= 4 +	"2 "	

Задача 1. Ряд натуральных чисел – трассировка(11)

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

i = 1;
do {
    printf("%d ", i);
    i = i + 1;
} while (i <= n);
```

i	n	i<=n	Вывод	Ввод
1	4		"n = "	4 <Enter>
2		2 <= 4 +	"1 "	
3		3 <= 4 +	"2 "	
			"3 "	

Задача 1. Ряд натуральных чисел – трассировка(12)

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

i = 1;
do {
    printf("%d ", i);
    i = i + 1;
} while (i <= n);
```

i	n	i<=n	Вывод	Ввод
1	4		"n = "	4 <Enter>
2		2 <= 4 +	"1 "	
3		3 <= 4 +	"2 "	
4			"3 "	

Задача 1. Ряд натуральных чисел – трассировка(13)

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

i = 1;
do {
    printf("%d ", i);
    i = i + 1;
} while (i <= n);
```

i	n	i<=n	Вывод	Ввод
1	4		"n = "	4 <Enter>
2		2 <= 4 +	"1 "	
3		3 <= 4 +	"2 "	
4		4 <= 4 +	"3 "	

Задача 1. Ряд натуральных чисел – трассировка(14)

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

i = 1;
do {
    printf("%d ", i);
    i = i + 1;
} while (i <= n);
```

i	n	i<=n	Вывод	Ввод
1	4		"n = "	4 <Enter>
2		2 <= 4 +	"1 "	
3		3 <= 4 +	"2 "	
4		4 <= 4 +	"3 "	
			"4 "	

Задача 1. Ряд натуральных чисел – трассировка(15)

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

i = 1;
do {
    printf("%d ", i);
    i = i + 1;
} while (i <= n);
```

i	n	i<=n	Вывод	Ввод
1	4		"n = "	4 <Enter>
2		2 <= 4 +	"1 "	
3		3 <= 4 +	"2 "	
4		4 <= 4 +	"3 "	
5			"4 "	

Задача 1. Ряд натуральных чисел – трассировка(16)

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

i = 1;
do {
    printf("%d ", i);
    i = i + 1;
} while (i <= n);
```

i	n	i<=n	Вывод	Ввод
1	4		"n = "	4 <Enter>
2		2 <= 4 +	"1 "	
3		3 <= 4 +	"2 "	
4		4 <= 4 +	"3 "	
5		5 <= 4 -	"4 "	

Задача 1. Ряд натуральных чисел

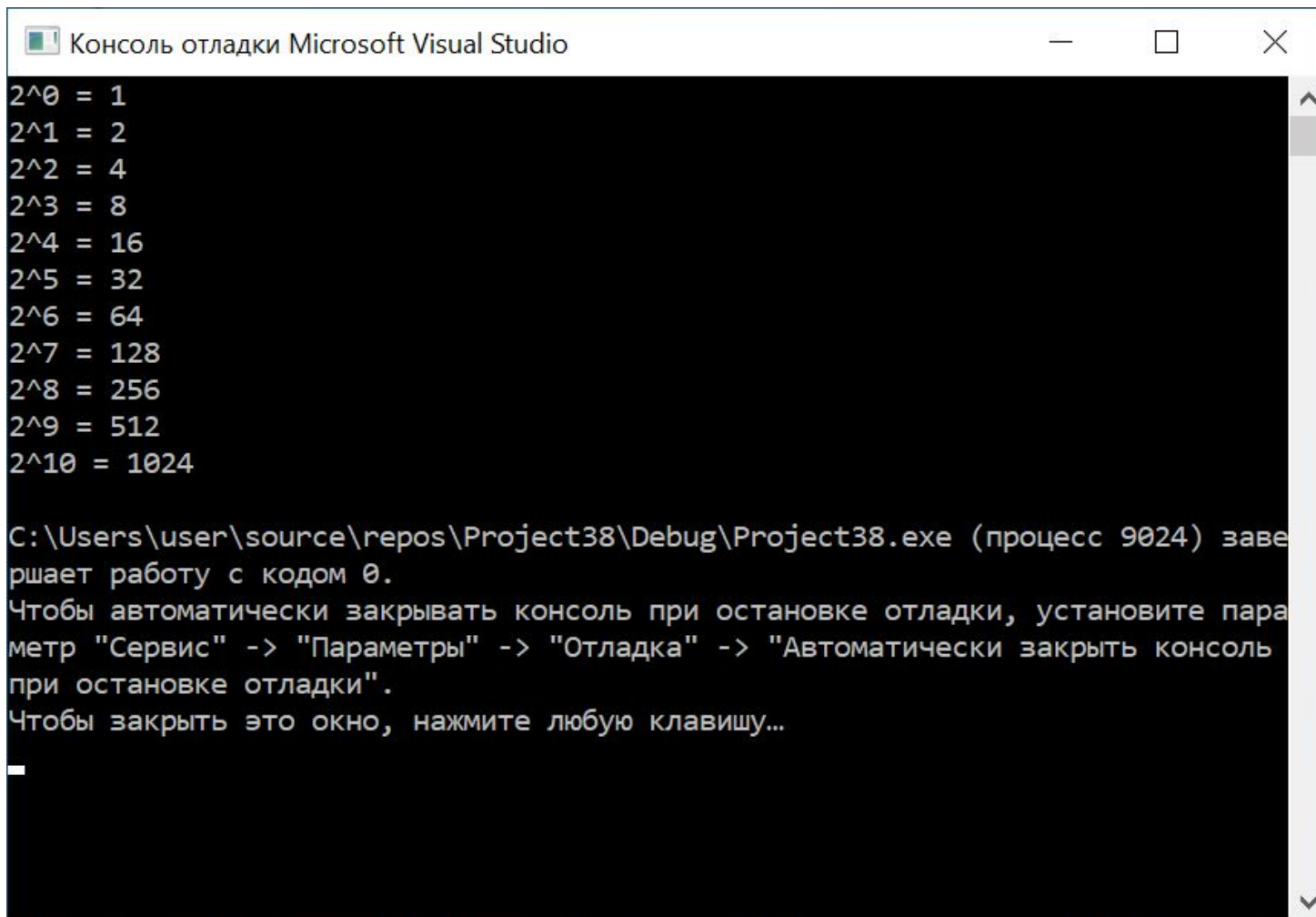
Вводится N .

Нужно вывести натуральные числа от 1 до N (включительно).

Нарисуйте блок схему к Задаче 1.

Задача 2. Степени двойки

Нужно вывести степени числа 2 от 0 до 10



```
Консоль отладки Microsoft Visual Studio

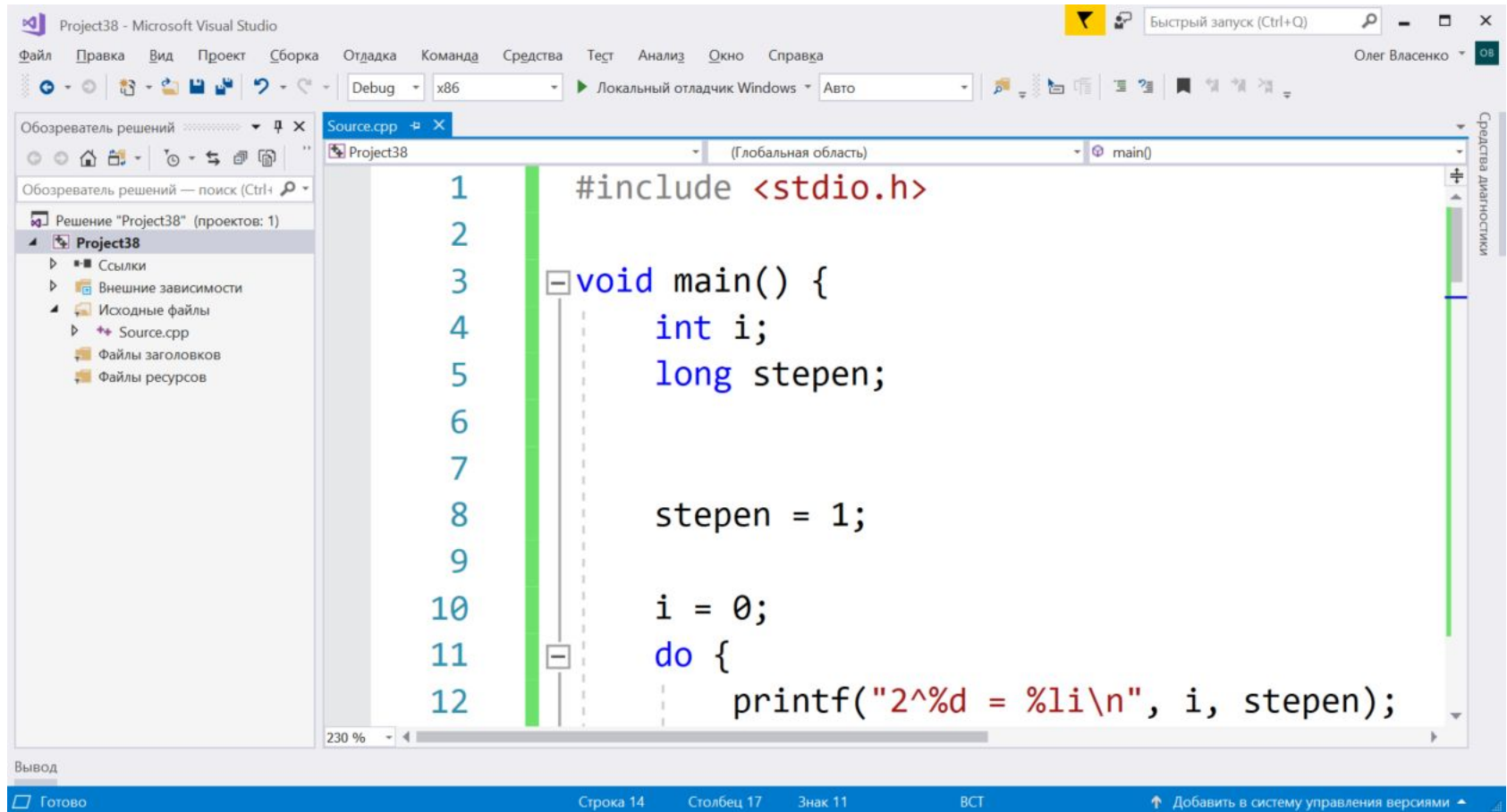
2^0 = 1
2^1 = 2
2^2 = 4
2^3 = 8
2^4 = 16
2^5 = 32
2^6 = 64
2^7 = 128
2^8 = 256
2^9 = 512
2^10 = 1024

C:\Users\user\source\repos\Project38\Debug\Project38.exe (процесс 9024) завершает работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, установите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Чтобы закрыть это окно, нажмите любую клавишу...
```

Задача 2. Степени двойки

Нужно вывести степени числа 2 от 0 до 10

(Ниже – начало программы)



The screenshot shows the Microsoft Visual Studio IDE with a C++ project named 'Project38'. The 'Source.cpp' file is open, displaying the following code:

```
1  #include <stdio.h>
2
3  void main() {
4      int i;
5      long stepen;
6
7
8      stepen = 1;
9
10     i = 0;
11     do {
12         printf("2^%d = %li\n", i, stepen);
```

The code is written in a C++ style, using `void main()` and `printf`. The variable `stepen` is declared as `long`. The program starts by including `<stdio.h>`, then defines `main()` where it initializes `stepen` to 1 and `i` to 0. It then enters a `do` loop, printing the current power of 2.

The IDE interface includes a menu bar (File, Edit, View, Project, Build, Debug, Command, Tools, Test, Analysis, Window, Help), a toolbar, and a sidebar with the 'Solution Explorer' showing the project structure. The status bar at the bottom indicates 'Готово' (Ready) and provides coordinates for the current cursor position: 'Строка 14' (Line 14), 'Столбец 17' (Column 17), and 'Знак 11' (Character 11).

Задача 2. Степени двойки - трассировка

Нужно вывести степени числа
2 от 0 до 5

```
void main() {  
    int i;  
    long stepen;  
  
    stepen = 1;  
  
    i = 0;  
    do {  
        printf("2^%d = %li\n", i, stepen);  
    }
```

stepen	i	i<=10	Вывод
1	0		"2^0 = 1"
2	1	1 <= 5 +	"2^1 = 2"
4	2	2 <= 5 +	"2^2 = 4"
8	3	3 <= 5 +	"2^3 = 8"
16	4	4 <= 5 +	"2^4 = 16"
32	5	5 <= 5 +	"2^5 = 32"
64	6	6 <= 5 -	

Задача 2. Степени двойки

Нужно вывести степени числа 2 от 0 до 10

Напишите код.

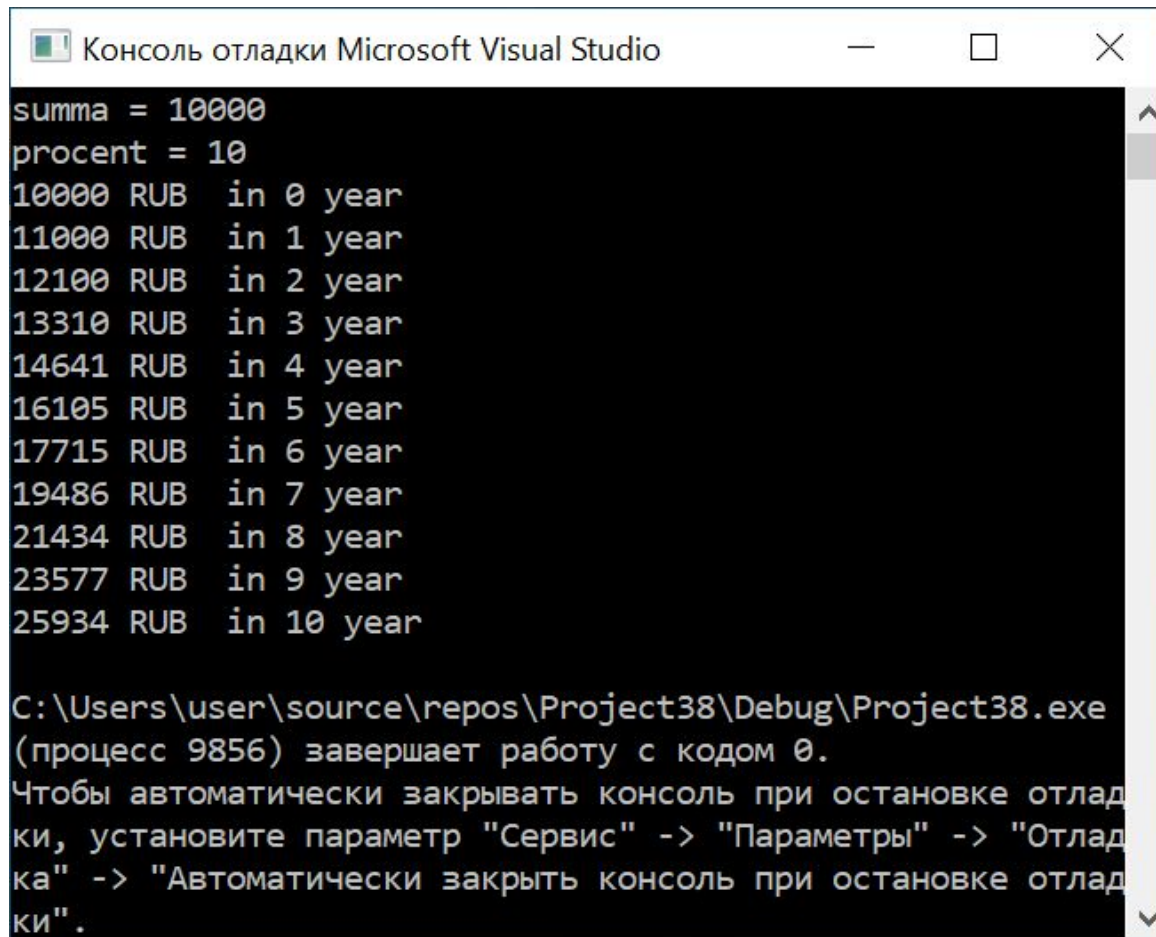
Нарисуйте блок схему к Задаче 2.

Задача 3*. Деньги на счете (сложные проценты)

На счет положили R рублей сегодня под N процентов годовых. Выведите какая будет сумма на счете через 1, 2, 3 ... 10 лет, если со счетом не будет никаких операций (с него не будут забирать деньги и не будут добавлять)

Задача 3*. Деньги на счете (сложные проценты)

На счет положили R рублей сегодня под N процентов годовых. Выведите какая будет сумма на счете через 1,2, 3 ... 10 лет, если со счетом не будет никаких операций (с него не будут забирать деньги и не будут добавлять)



```
Консоль отладки Microsoft Visual Studio

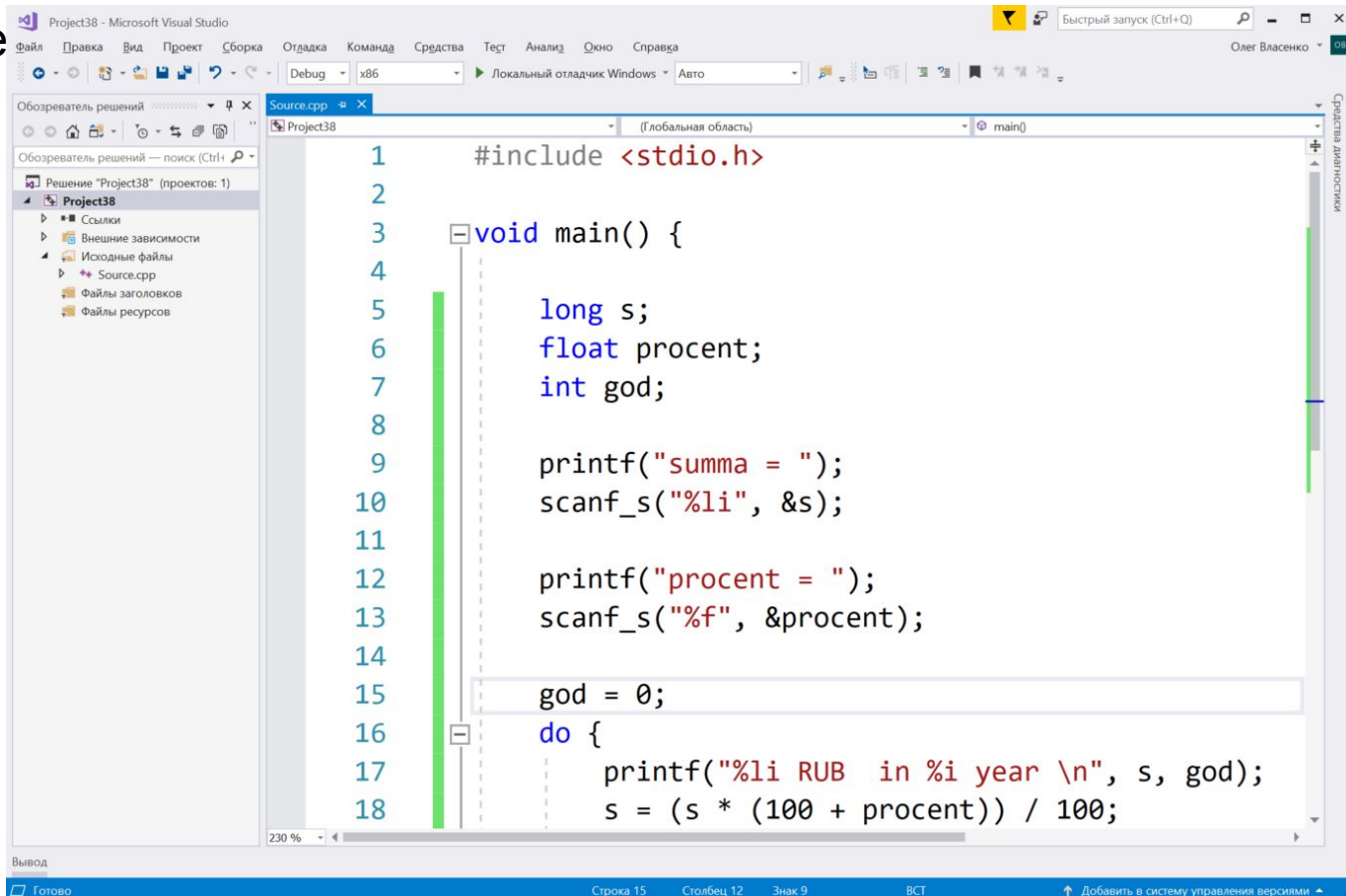
summa = 10000
procent = 10
10000 RUB in 0 year
11000 RUB in 1 year
12100 RUB in 2 year
13310 RUB in 3 year
14641 RUB in 4 year
16105 RUB in 5 year
17715 RUB in 6 year
19486 RUB in 7 year
21434 RUB in 8 year
23577 RUB in 9 year
25934 RUB in 10 year

C:\Users\user\source\repos\Project38\Debug\Project38.exe
(процесс 9856) завершает работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, установите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".
```


Задача 3*. Деньги на счете (сложные проценты)

На счет положили R рублей сегодня под N процентов годовых. Выведите какая будет сумма на счете через 1,2, 3 ... 10 лет, если со счетом не будет никаких операций (с него не будут забирать деньги и не будут добавлять)

(Ниж



The screenshot shows the Microsoft Visual Studio IDE with a C++ project named "Project38". The "Source.cpp" file is open, displaying the following code:

```
1  #include <stdio.h>
2
3  void main() {
4
5      long s;
6      float procent;
7      int god;
8
9      printf("summa = ");
10     scanf_s("%li", &s);
11
12     printf("procent = ");
13     scanf_s("%f", &procent);
14
15     god = 0;
16     do {
17         printf("%li RUB in %i year \n", s, god);
18         s = (s * (100 + procent)) / 100;
```

The code is partially visible, showing the beginning of the `main` function. It includes headers, declares variables for sum (`s`), interest rate (`procent`), and years (`god`), and starts the input process for sum and interest rate. The loop body for calculating the sum for each year is partially visible.

Задача 3*. Деньги на счете (сложные проценты)

На счет положили R рублей сегодня под N процентов годовых.
Выведите какая будет сумма на счете через 1, 2, 3 ... 10 лет, если со счетом не будет никаких операций (с него не будут забирать деньги и не будут добавлять)

Нарисуйте блок схему к Задаче 3.

Домашнее задание

1. Доделать дома задачи 1-2 – если они не были доделаны в классе.
2. Дорисовать блок схемы к задачам 1-2 – если они не были дорисованы или были с ошибками.

Задача 4. Реализовать задачу «Вводится число N. Нужно вывести первые N четных чисел». Нарисовать блок схему. Подготовить необходимые тесты.

Пример теста:

ВВОД N = 6

ВЫВОД 2 4 6 8 10 12

Задача 5.1. Придумать задачу на последовательность чисел («вывести все нечетные числа от 1 до N (1 3 5 7 9 ...)», «вывести степени числа 4 от 0 до N (1 4 16 64 ...)», «Вывести первые N чисел Фибоначчи (0 1 1 2 3 5 8 ...)» и т.п.) и реализовать её используя DO WHILE. Для вашей реализации нарисовать блок-схему. Прописать необходимые тесты.

ИЛИ

Задача 5.2. Придумать задачу из реальной жизни, экономики, финансов («Подсчитать накопленную инфляцию за N лет, если она все годы одинаковая и составляет K % в год», «Каждый месяц студент Вася откладывает 1000 рублей про запас. Вывести его запасы в ближайшие 36 месяцев (1000 2000 3000 4000 ...)» и т.п.). Реализовать задачу используя DO WHILE – код, блок-схема, необходимые тесты.

В итоге у всех будут выполнены задачи 1, 2, 4 и одна из задач 5.1 или 5.2. У самых быстрых будет выполнена еще и задача 3*.

Домашнее задание – оформление

Для сдачи домашней работы нужно иметь для каждой задачи:

1) код программы

2) Блок схему, нарисованную от руки (можно под линейку, можно без линейки)

3) Тесты

Срок выполнения – до следующей встречи на лабораторной работе

Если болел/не мог – это становится «долгом»