Домашняя работа №1

построение таблицы трассировки для ветвящегося процесса

Вложенные операторы условия : домашняя работа №1

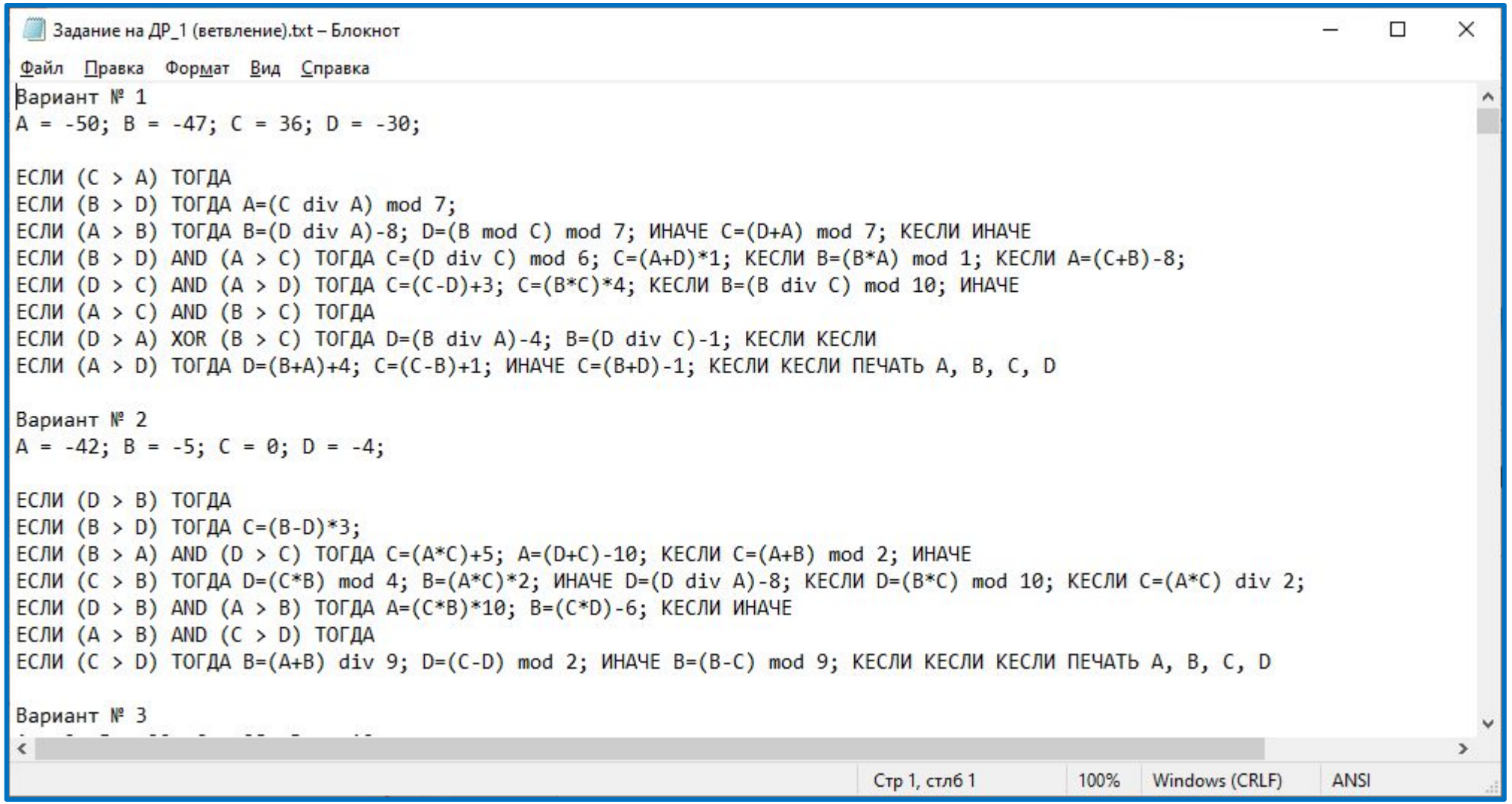
Задание:

Для заданного варианта (номер варианта – ***две последние цифры шифра студента*** (номер студенческого билета))

- построить блок-схему и программу на языке Си (C++) (с использованием программы **afce .exe**);
- осуществить запуск полученной программы в среде программирования;
- осуществить пошаговое ее выполнение, протоколируя в таблице трассировке промежуточные результаты.

Пример задания

Файл: ***Задание на ДР_1 (ветвление).txt***



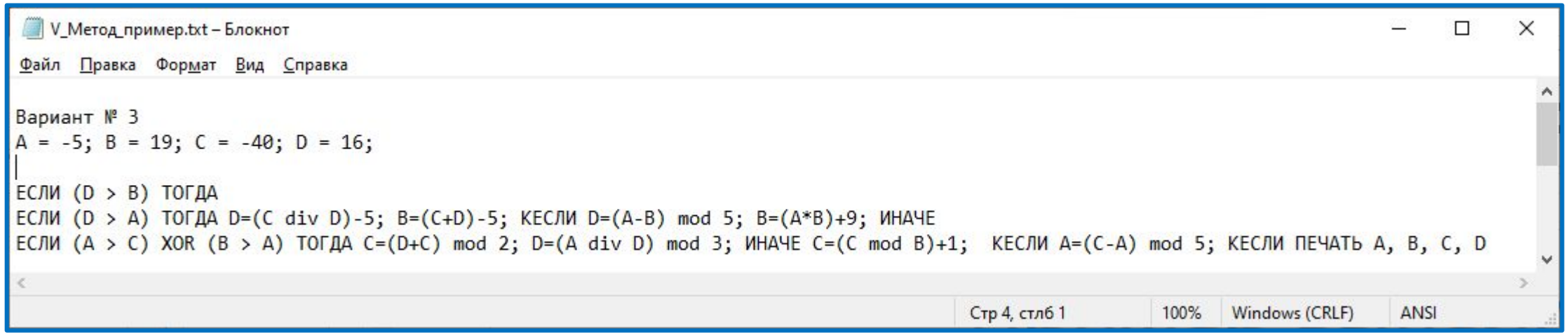
Методика выполнения

Задание представлено в виде неструктурированного кода на алгоритмическом языке. В ходе выполнения задание необходимо:

1. Открыть файл задания и найти свой вариант. Скопировать задание в отчёт (doc-файл).
2. Сделать дополнительную копию задания в отчете. Представить задание в виде структурного кода, отступом отразив вложенность конструкций.
3. Запустить программу редактора блок-схем (**afce.exe**), прилагаемую вместе с заданием.
   - Выбрать язык программирования Си.
   - Построить блок-схему в соответствии с заданием на шаге 2
   - Сохранить изображение блок-схемы как растровую картинку.
   - Вставить растровое изображение в отчет ***(или на шаге 4 с номерами блоков)***
4. Пронумеровать каждый блок блок-схемы.
5. Из редактора блок-схем скопировать исходный код в среду программирования. Дополнить программу в соответствии с синтаксисом языка программирования Си (C++).
6. Запустить программу в среде программирования, сделать скриншот исходника и результатов запуска, вставить полученное изображение в отчет.
7. Построить в отчете таблицу трассировки.
8. Сравнить результат таблицы трассировки с результатом работы программы.

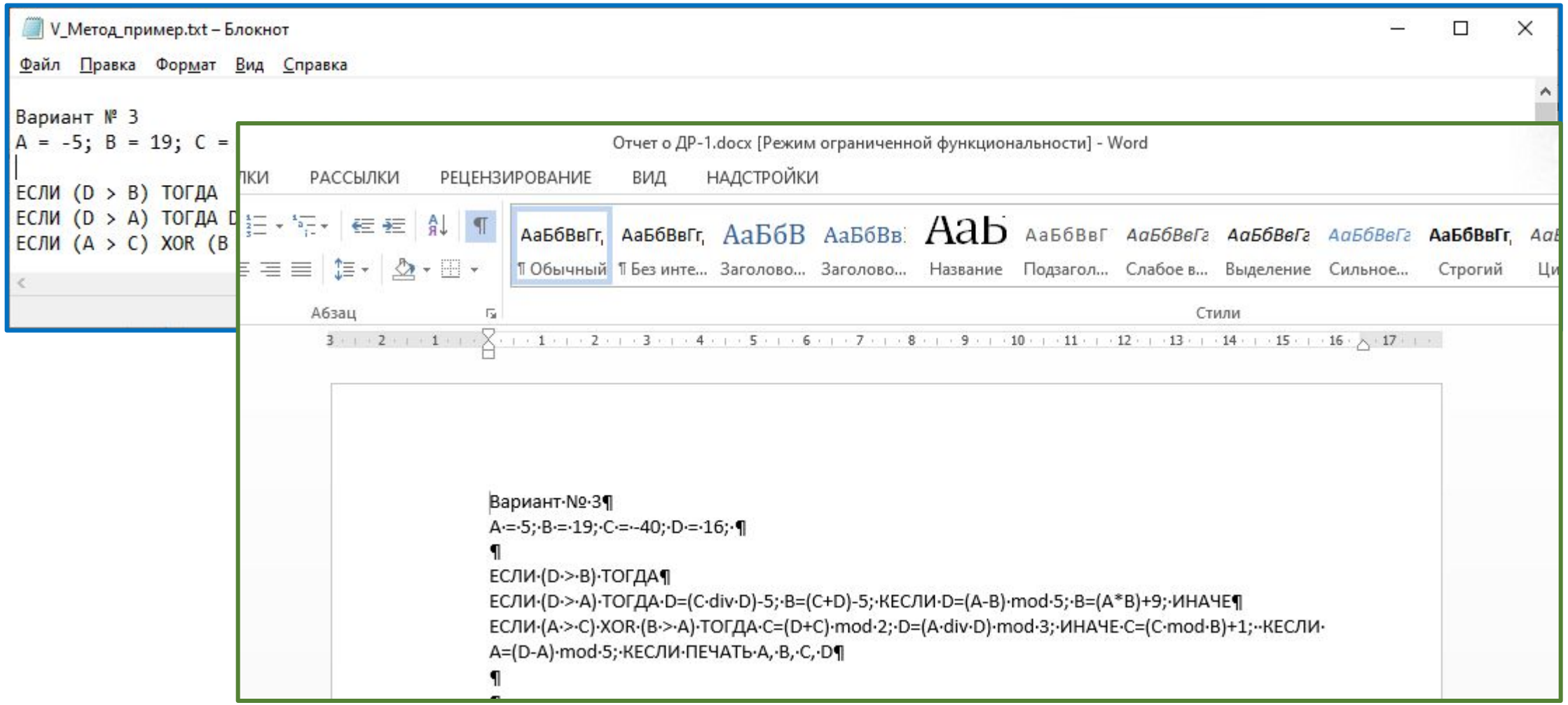
Пример выполнения (шаг 1)

1. Открыть файл задания и найти свой вариант. Скопировать задание в отчёт (doc-файл).



Пример выполнения (шаг 1)

1. Открыть файл задания и найти свой вариант. Скопировать задание в отчёт (doc-файл).



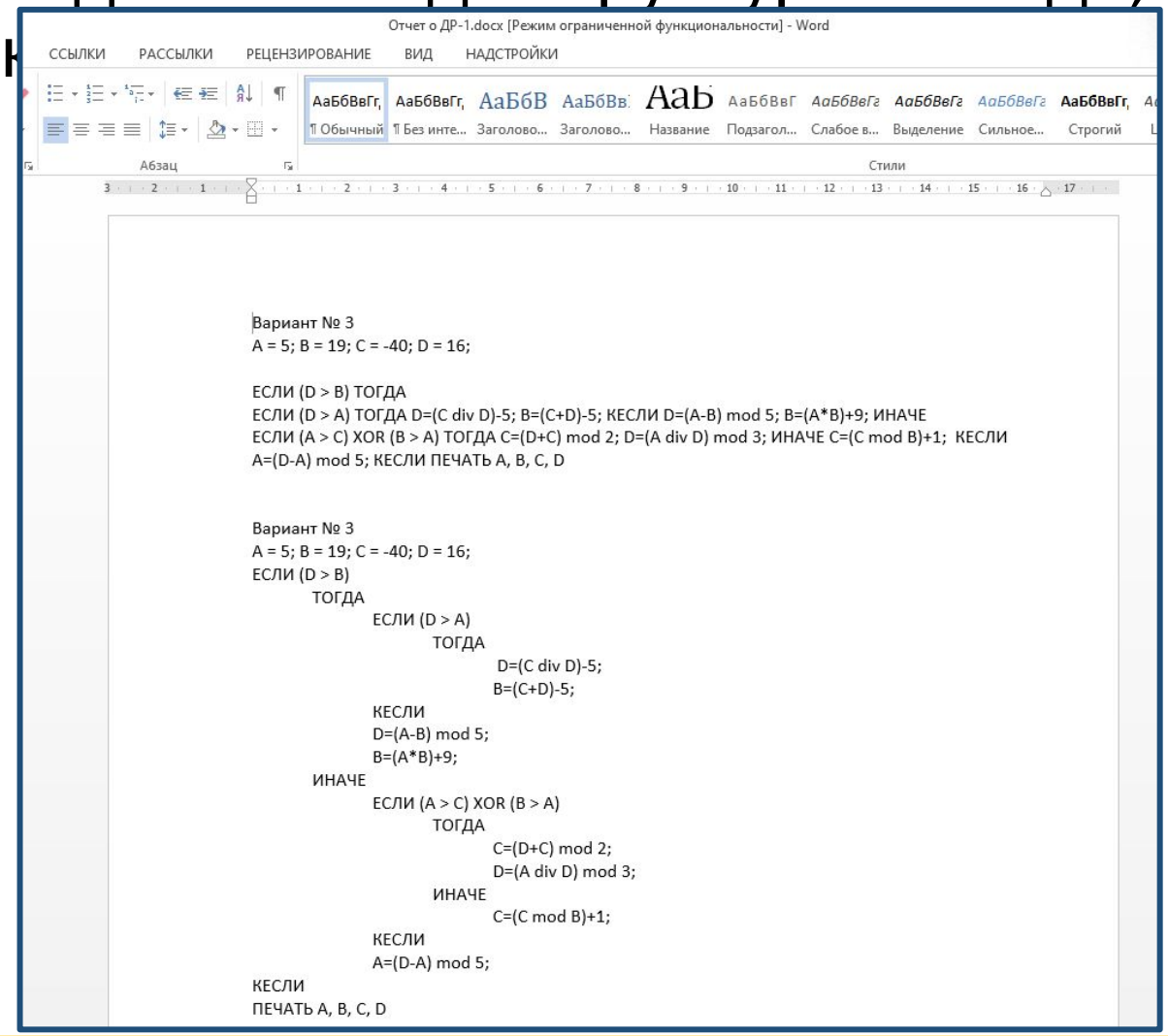
Пример выполнения (шаг 2)

2. Сделать дополнительную копию задания в отчете. Представить задание в виде структурного кода, отступом отразив вложенность конструкций.

(… открыт блокнот)

Пример выполнения (шаг 2)

2. Сделать дополнительную копию задания в отчете. Представить задание в виде структурного кода, отступом отразив вложенность к



Вариант № 3
A = 5; B = 19; C = -40; D = 16;

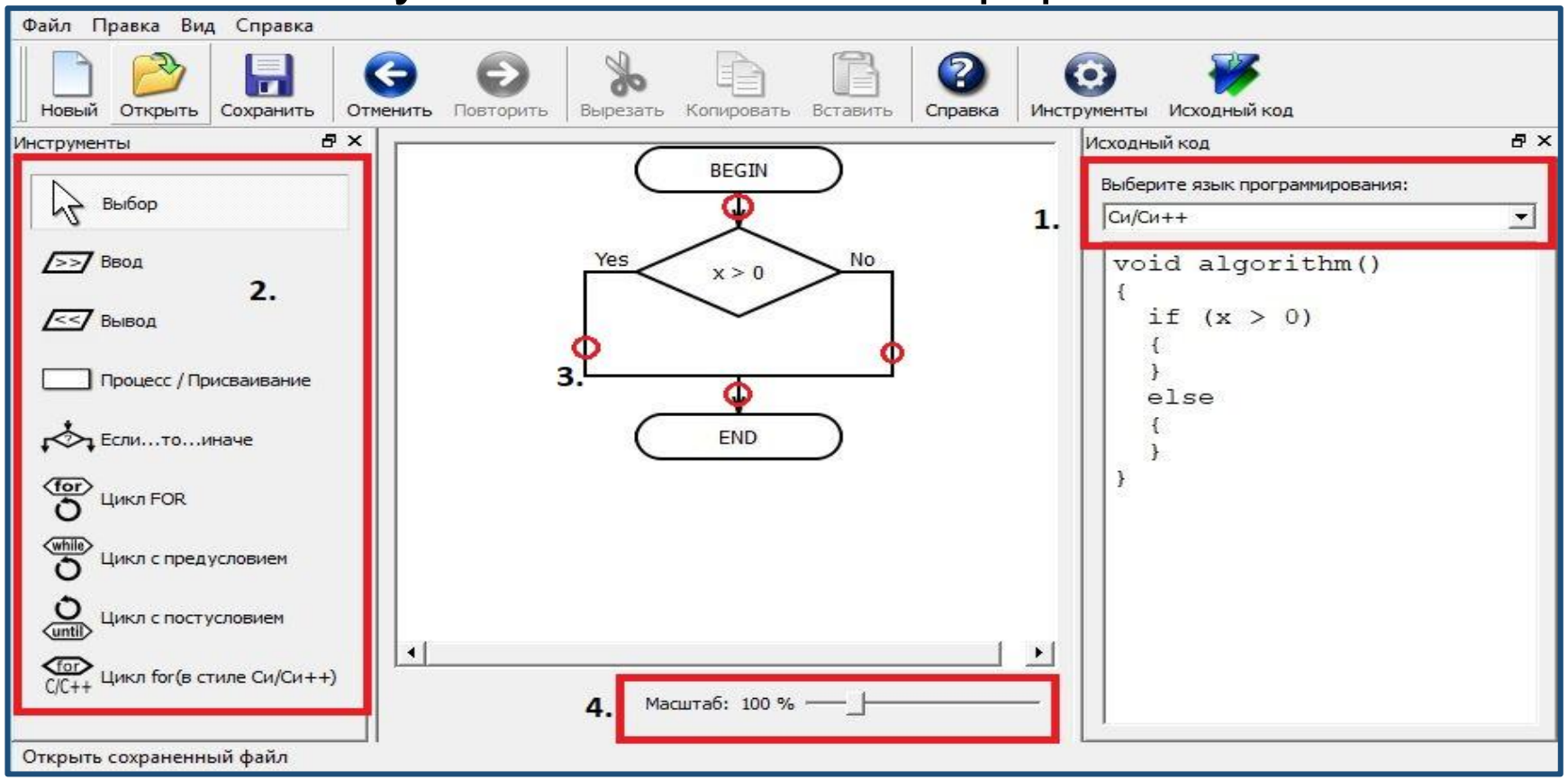
ЕСЛИ (D > B) ТОГДА
ЕСЛИ (D > A) ТОГДА D=(C div D)-5; B=(C+D)-5; КЕСЛИ D=(A-B) mod 5; B=(A*B)+9; ИНАЧЕ
ЕСЛИ (A > C) XOR (B > A) ТОГДА C=(D+C) mod 2; D=(A div D) mod 3; ИНАЧЕ C=(C mod B)+1; КЕСЛИ
A=(D-A) mod 5; КЕСЛИ ПЕЧАТЬ A, B, C, D

```
Вариант № 3
A = 5; B = 19; C = -40; D = 16;
ЕСЛИ (D > B)
        ТОГДА
                ЕСЛИ (D > A)
                        ТОГДА
                                D=(C div D)-5;
                                B=(C+D)-5;
                КЕСЛИ
                D=(A-B) mod 5;
                B=(A*B)+9;
        ИНАЧЕ
                ЕСЛИ (A > C) XOR (B > A)
                        ТОГДА
                                C=(D+C) mod 2;
                                D=(A div D) mod 3;
                        ИНАЧЕ
                                C=(C mod B)+1;
                КЕСЛИ
                A=(D-A) mod 5;
КЕСЛИ
ПЕЧАТЬ A, B, C, D
```

```
Вариант № 3
A = 5;
B = 19;
C = -40;
D = 16;
ЕСЛИ (D > B)
        ТОГДА
                ЕСЛИ (D > A)
                        ТОГДА
                                D=(C div D)-5;
                                B=(C+D)-5;
                КЕСЛИ
                D=(A-B) mod 5;
                B=(A*B)+9;
        ИНАЧЕ
                ЕСЛИ (A > C) XOR (B > A)
                        ТОГДА
                                C=(D+C) mod 2;
                                D=(A div D) mod 3;
                        ИНАЧЕ
                                C=(C mod B)+1;
                КЕСЛИ
                A=(D-A) mod 5;
КЕСЛИ
ПЕЧАТЬ A, B, C, D
```

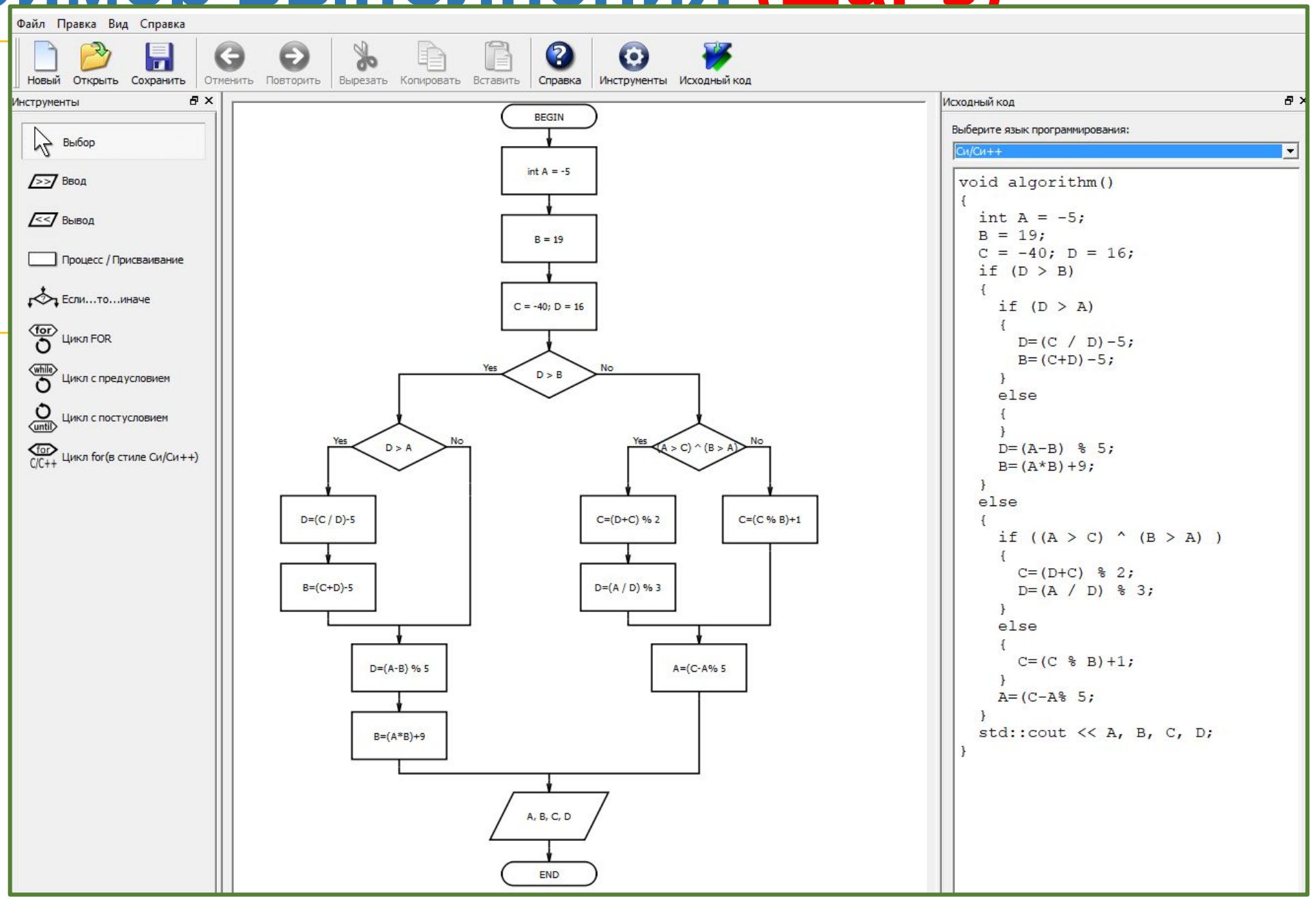
Пример выполнения (шаг 3)

3. Запустить программу редактора блок-схем, прилагаемую вместе с заданием. Выбрать нужный язык программирования (1). Затем начать построение блок-схемы в соответствии с заданием, выбирая нужные элементы (2) и выбирая точки их расположения (3). После окончания построения выбрать масштаб (4) так, чтобы блок-схема уместилась на листе формата А4.



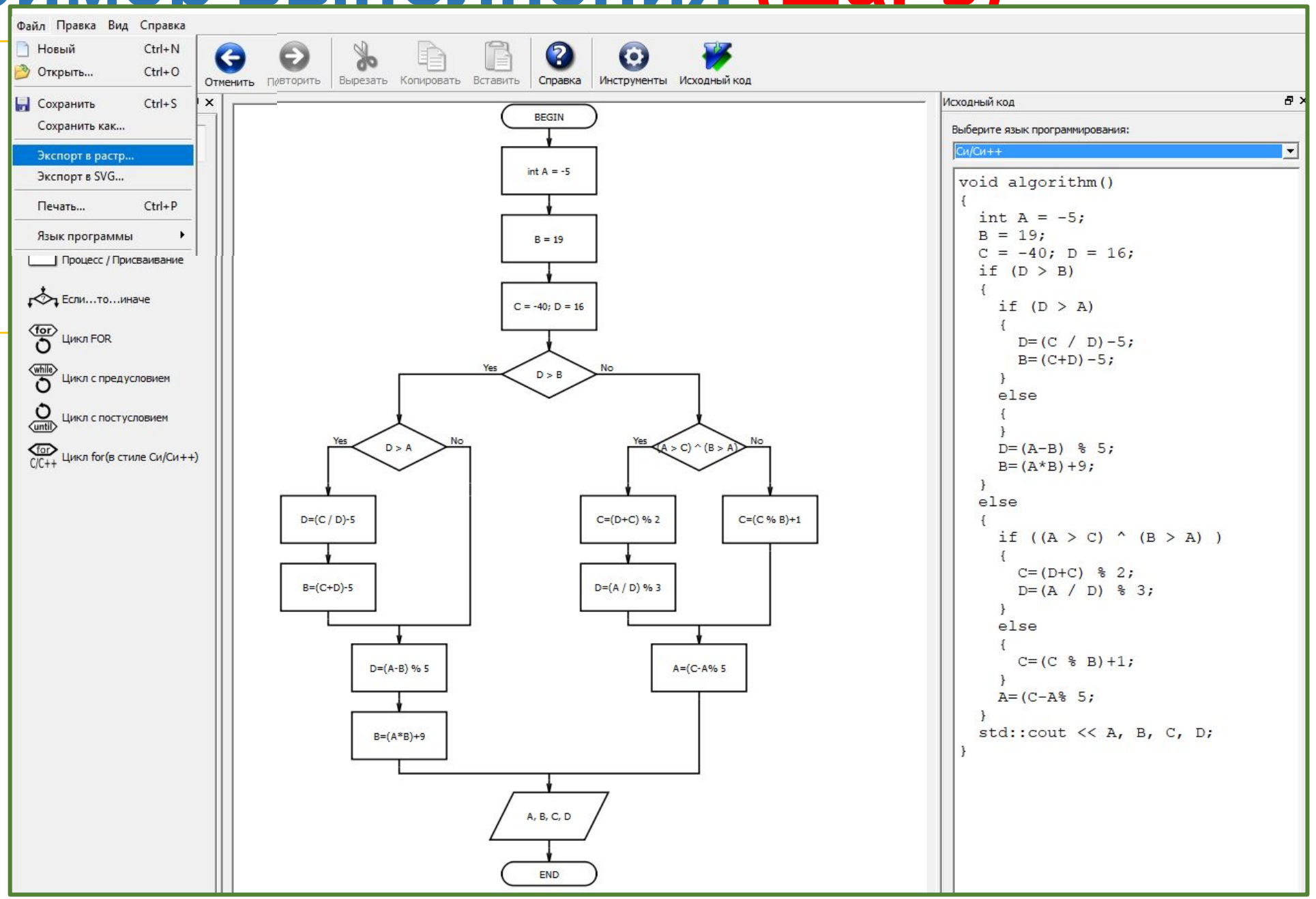
Пример выполнения (шаг 3)

3. Результат построения блок-схемы:



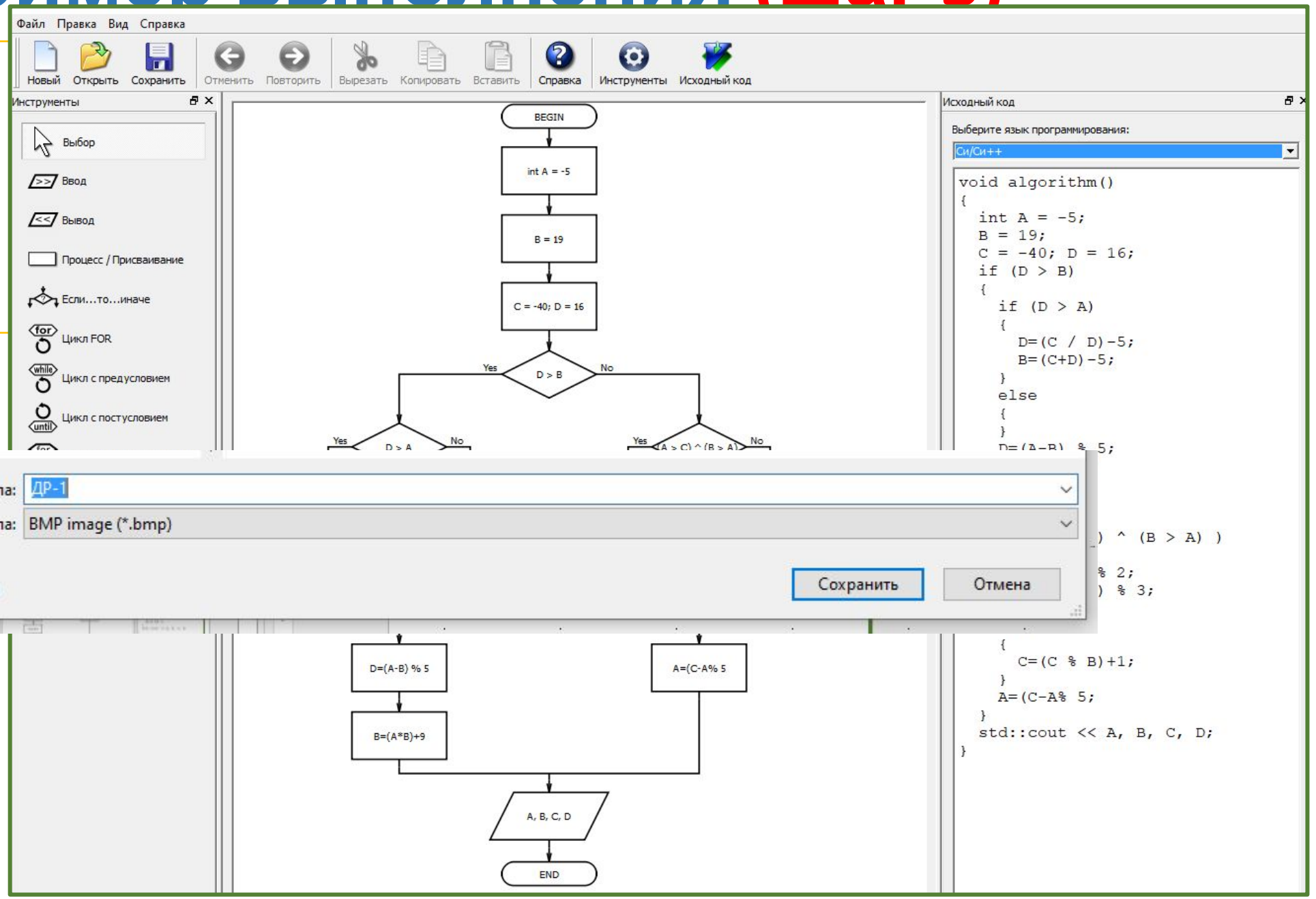
Пример выполнения (шаг 3)

3. Результат построения блок-схемы:



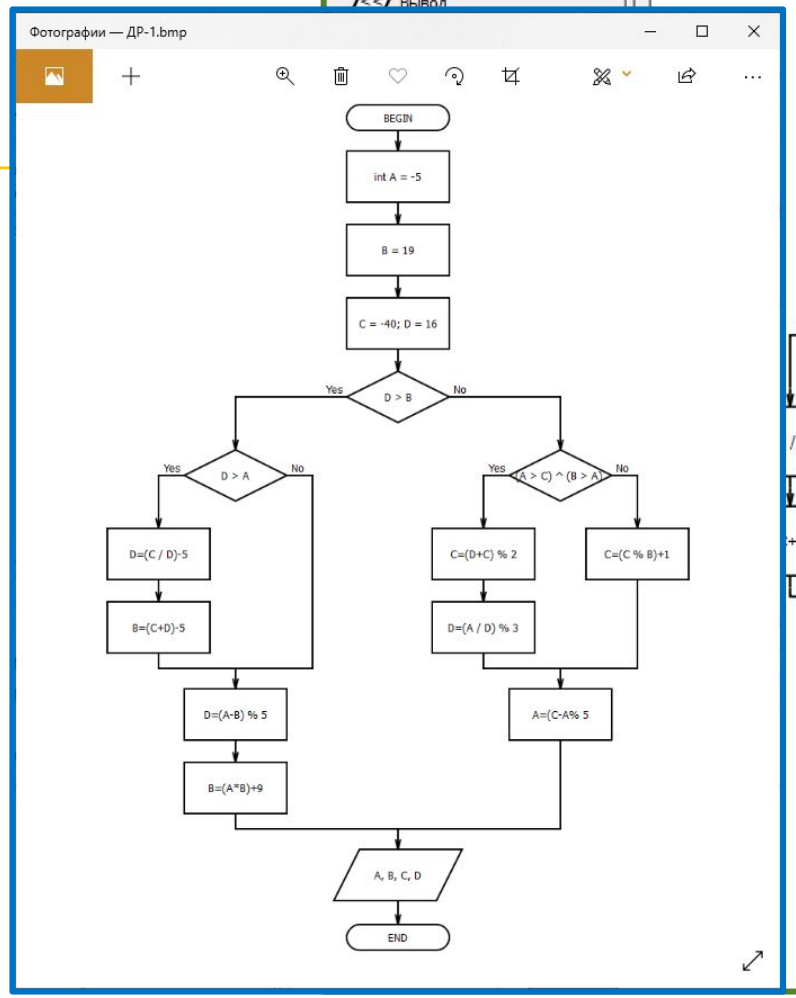
Пример выполнения (шаг 3)

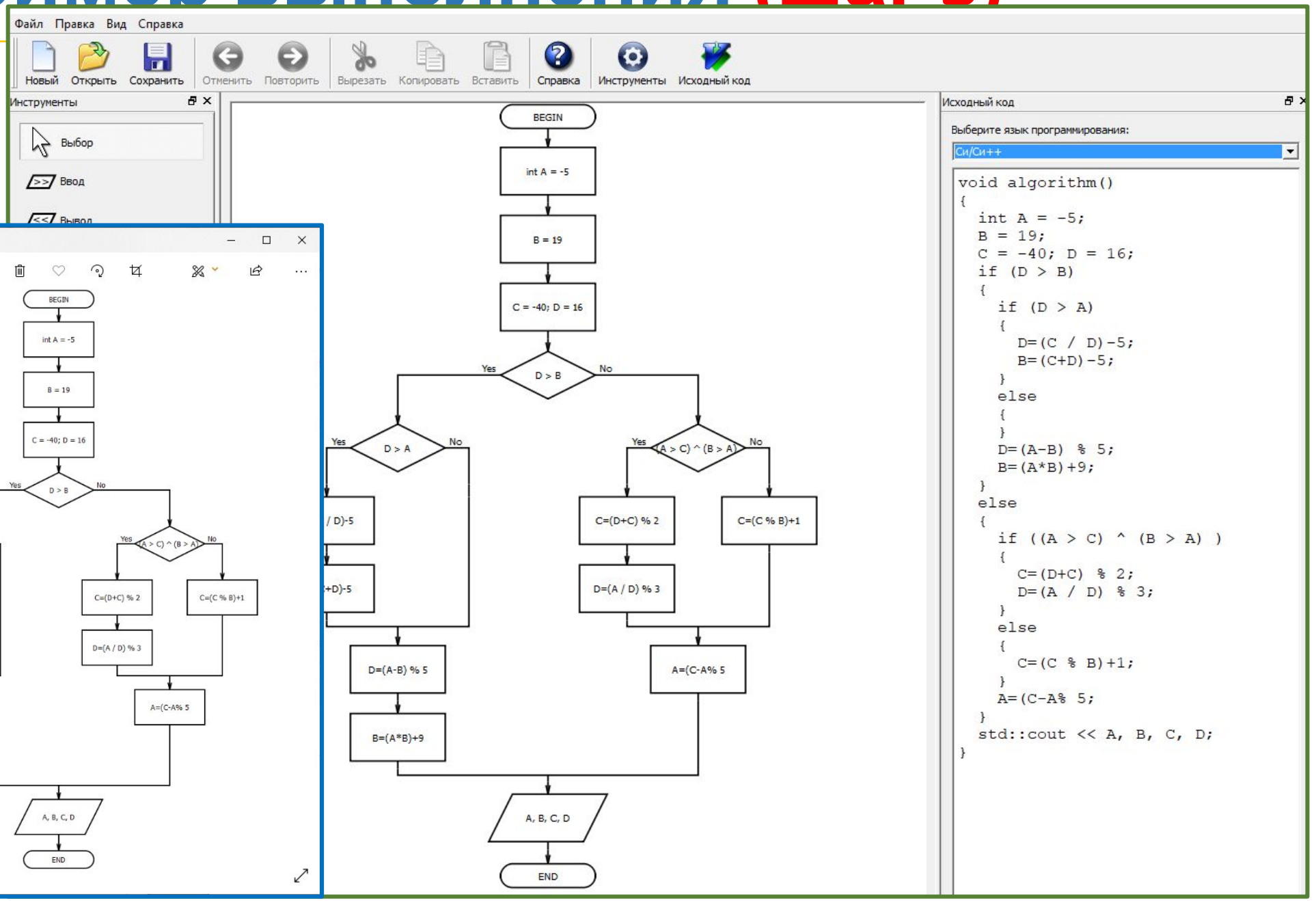
3. Результат построения блок-схемы:



Пример выполнения (шаг 3)

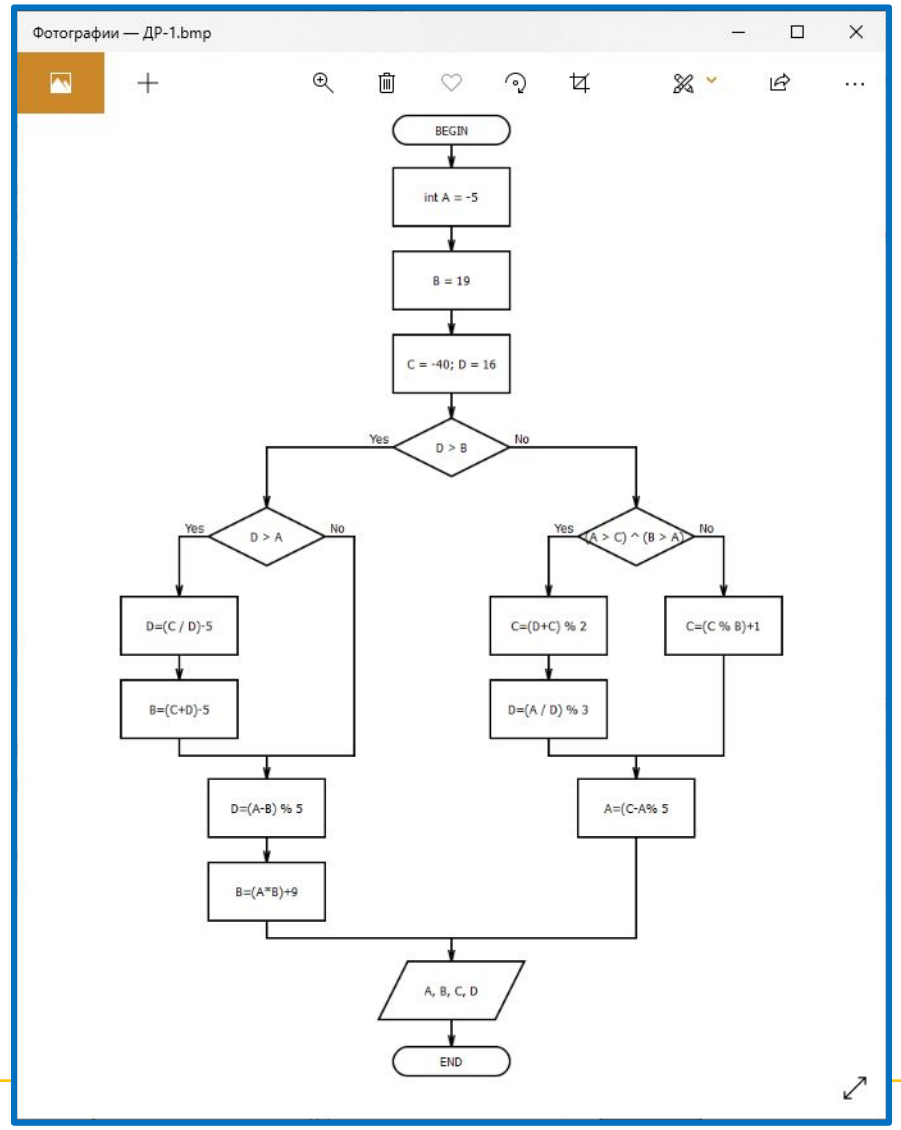
3. Вставить растровое изображение в отчет

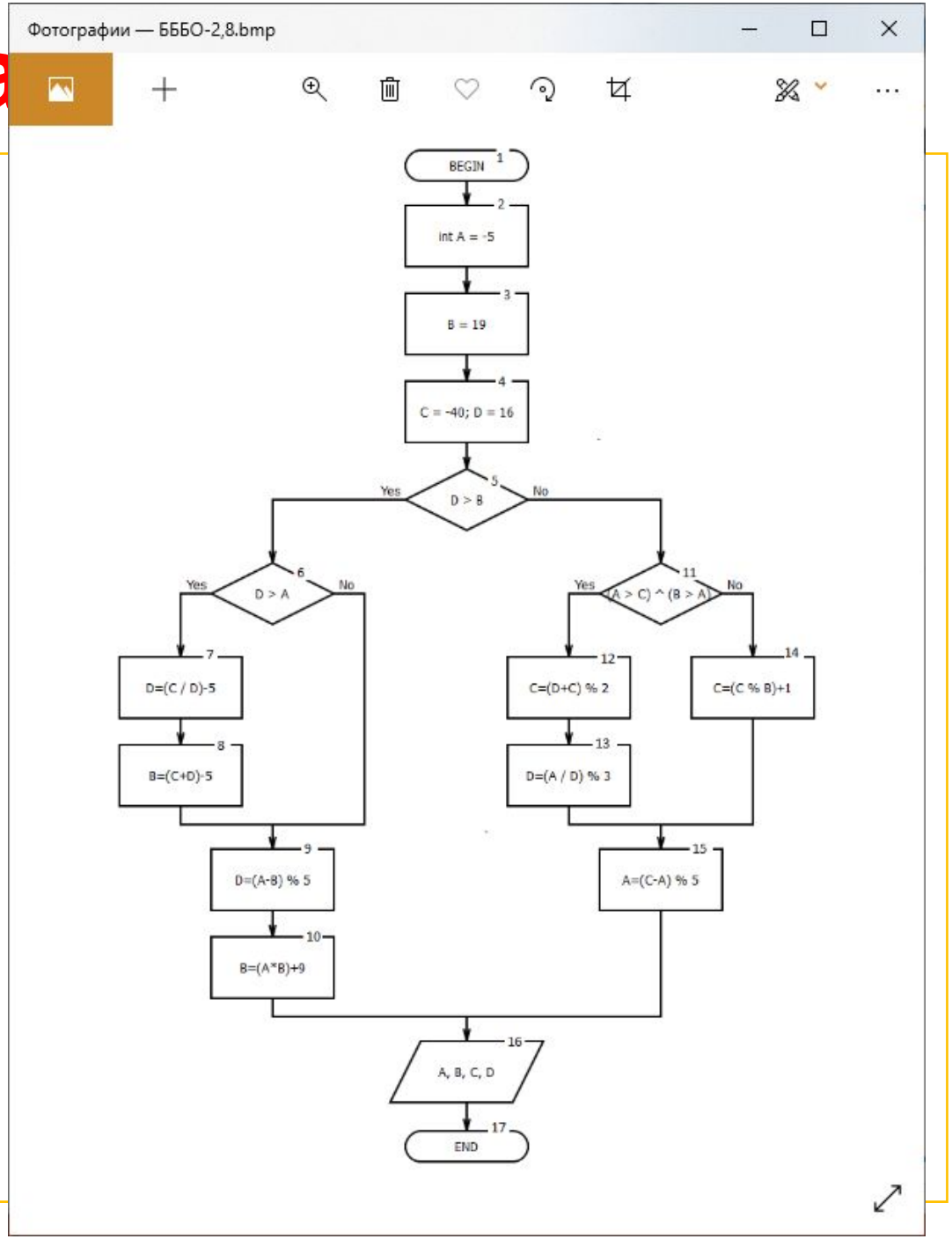




Пример выполнения (ша

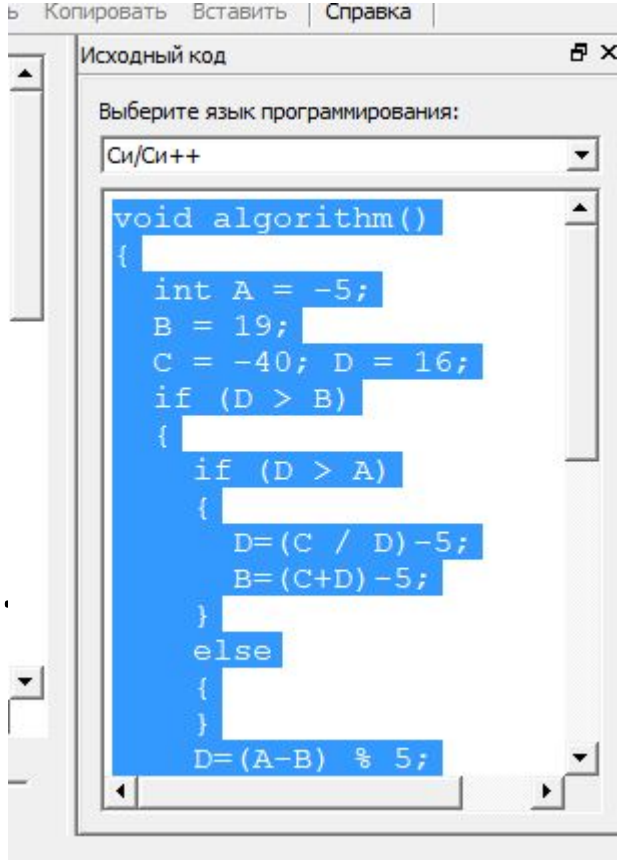
4. Пронумеровать каждый блок блок-схемы.

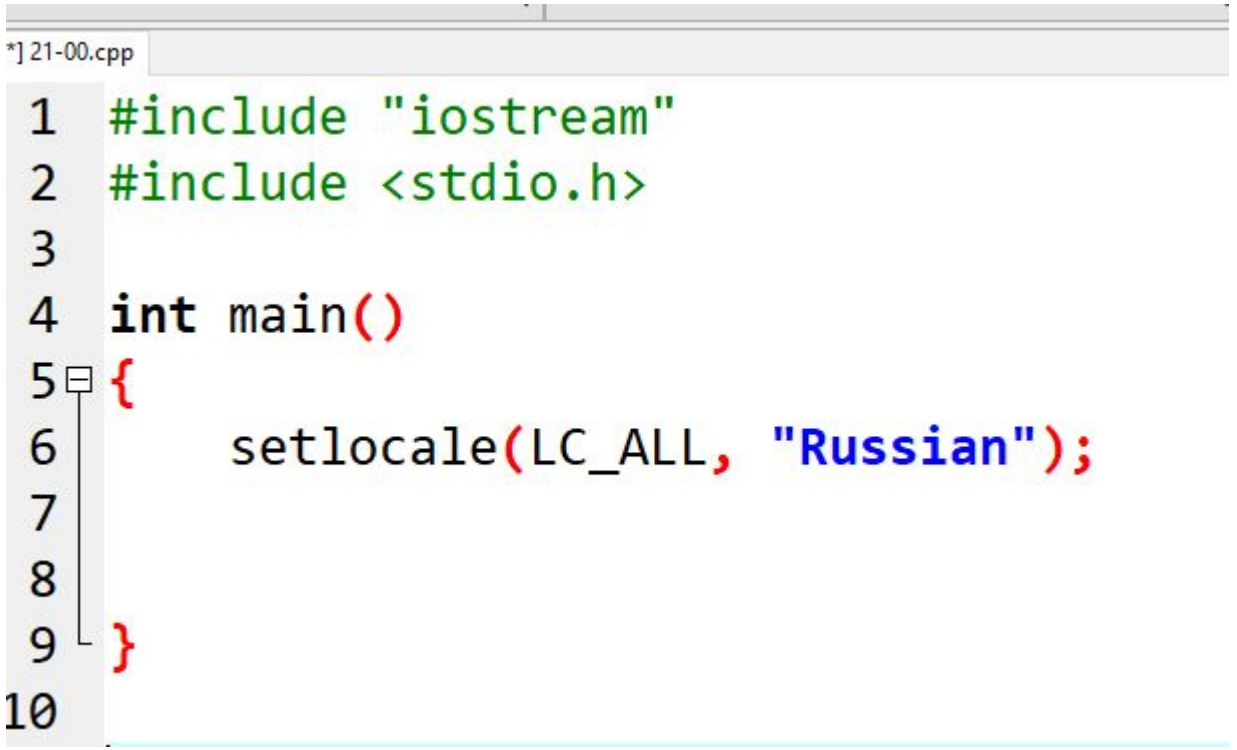




Пример выполнения (шаг 5)

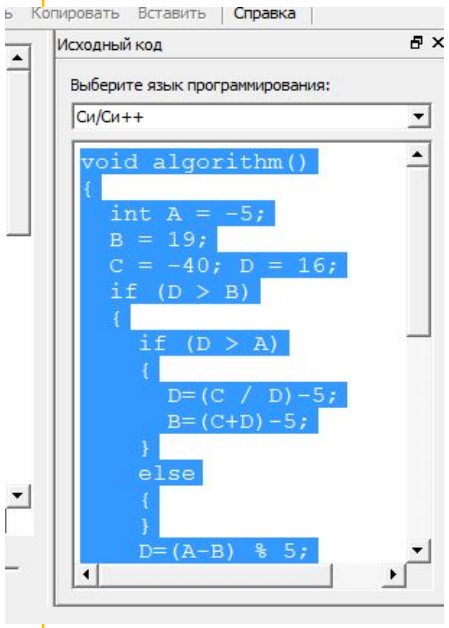
5. Из редактора блок-схем скопировать исходный код в среду программирования. Дополнить программу в соответствии с синтаксисом языка программирования Си (C++).

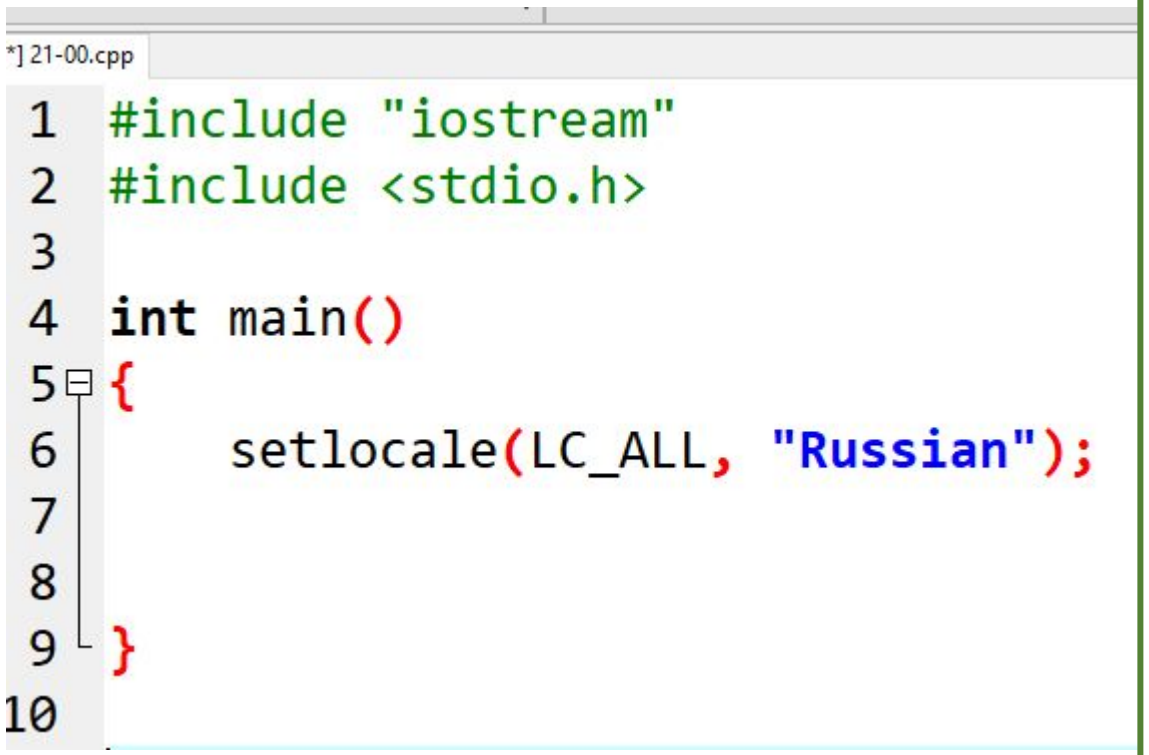


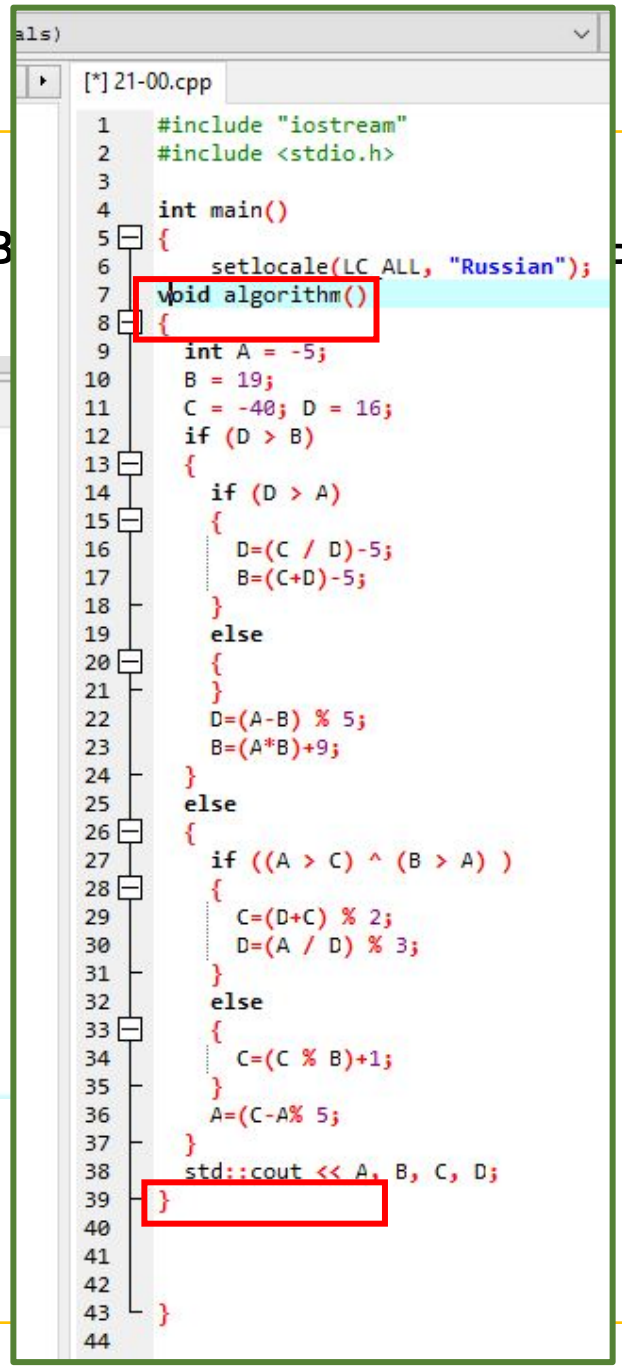


Пример выполнения (шаг 5)

5. Из редактора блок-схем скопировать исходный код в
программирования. Дополнить программу в соответств ыка
программирования Си (C++).

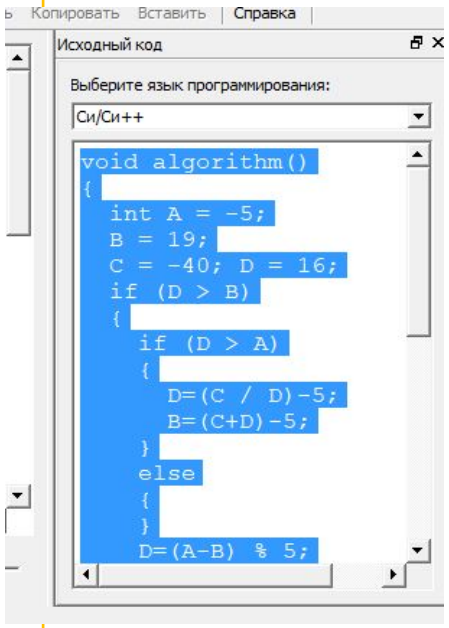


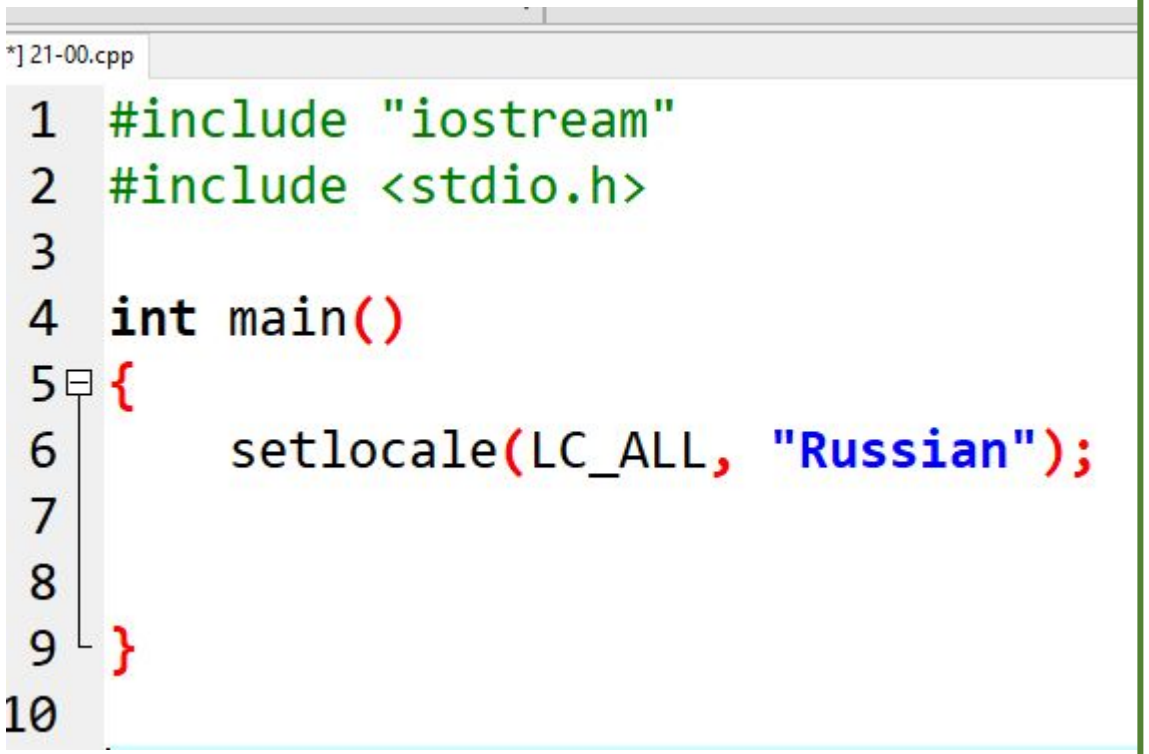


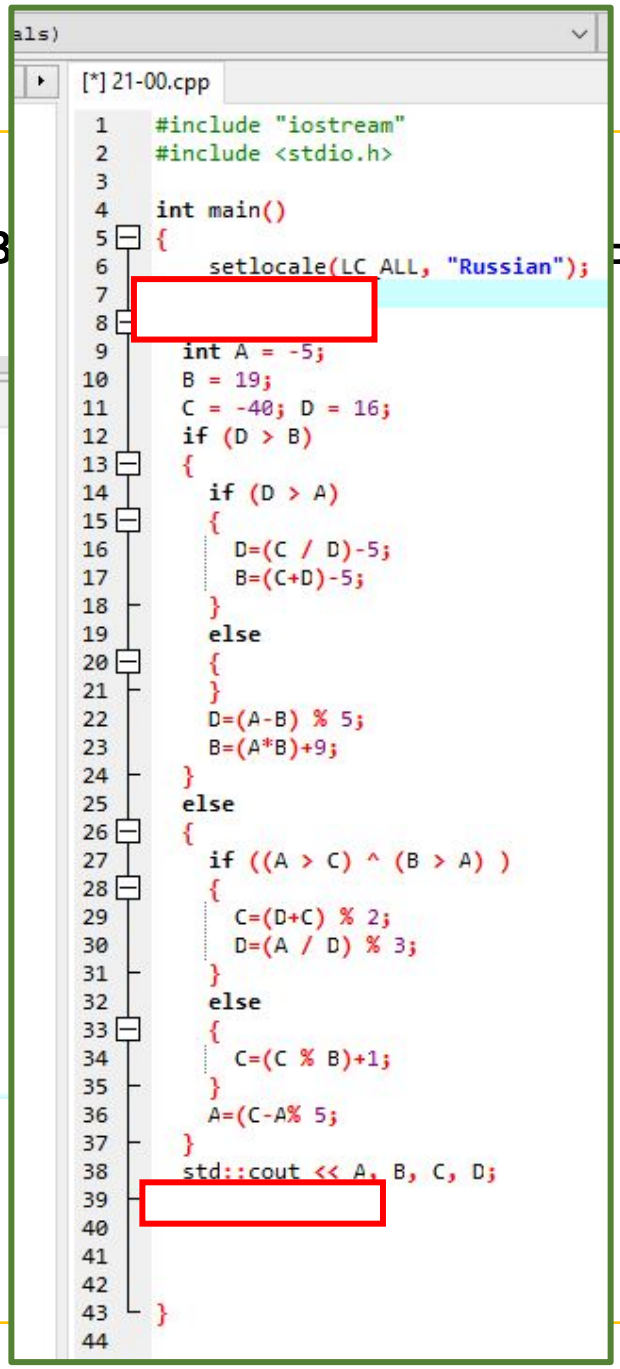


Пример выполнения (шаг 5)

5. Из редактора блок-схем скопировать исходный код в
программирования. Дополнить программу в соответств
программирования Си (C++).

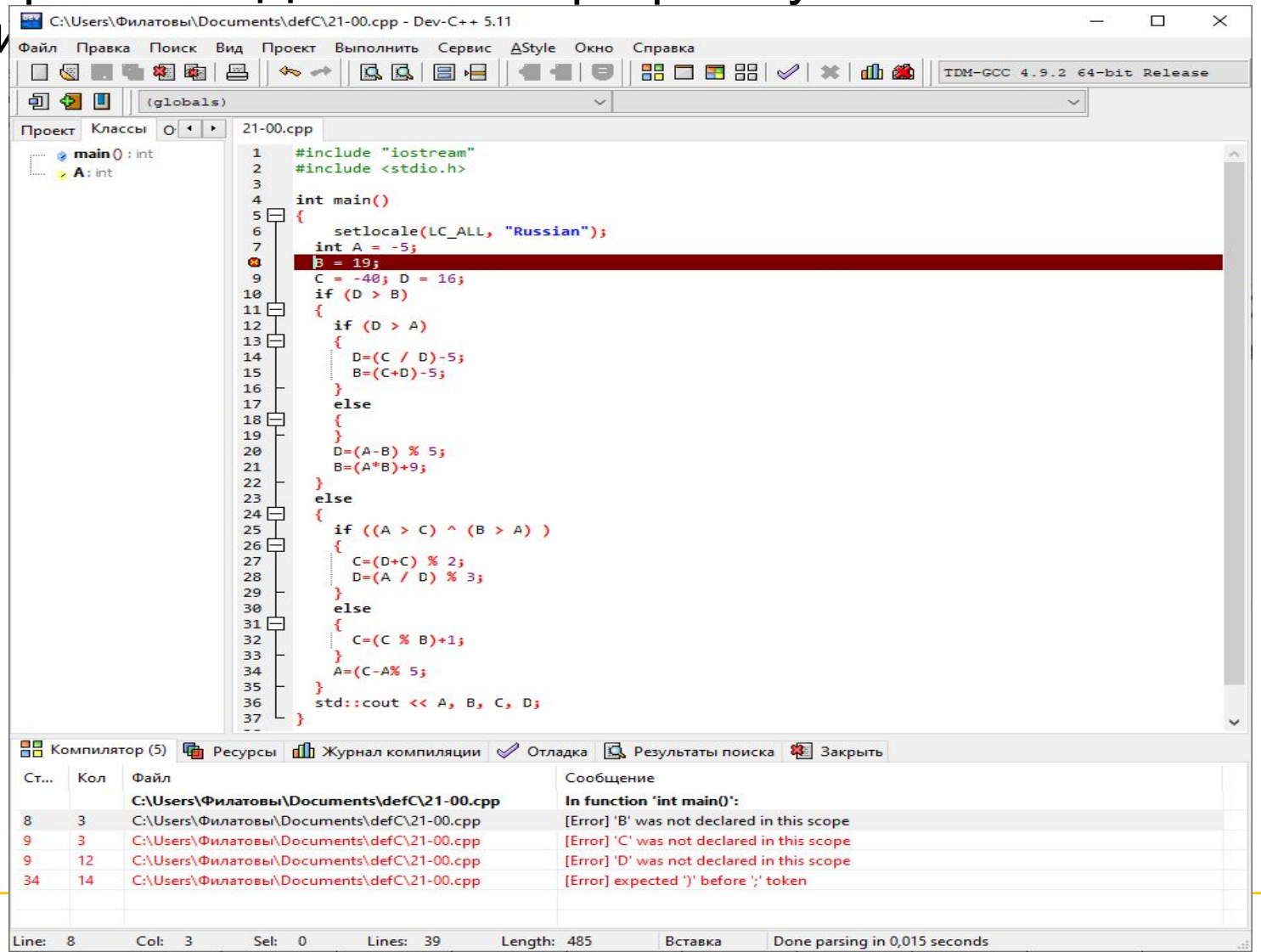






Пример выполнения (шаг 5)

5. Из редактора блок-схем скопировать исходный код в среду программирования. Дополнить программу в соответствии с синтаксисом языка программи



Пример выполнения (шаг 5)

5. Из реда код в среду
программ ветствии с синтаксисом языка
программ

21-00.cpp

```
1  #include "iostream"
2  #include <stdio.h>
3
4  int main()
5  {
6      setlocale(LC_ALL, "Russian");
7    int A = -5;
8    int B = 19;
9    int C = -40, D = 16;
10   if (D > B)
11   {
12     if (D > A)
13     {
14       D=(C / D)-5;
15       B=(C+D)-5;
16     }
17     else
18     {
19     }
20     D=(A-B) % 5;
21     B=(A*B)+9;
22   }
23   else
24   {
25     if ((A > C) ^ (B > A) )
26     {
27       C=(D+C) % 2;
28       D=(A / D) % 3;
29     }
30     else
31     {
32       C=(C % B)+1;
33     }
34     A=(C-A)% 5;
35   }
36   std::cout << A, B, C, D;
37 }
```

курсы | Журнал компиляции | Отладка | Результаты поиска | Закрыть

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
```

Пример выполнения (шаг 5)

5. Из реда код в среду
программ ветствии с синтаксисом языка
программ

21-00.cpp

```
1   #include "iostream"
2   #include <stdio.h>
3
4   int main()
5   {
6       setlocale(LC_ALL, "Russian");
7     int A = -5;
8     int B = 19;
9     int C = -40, D = 16;
10    if (D > B)
11    {
12      if (D > A)
13      {
14        D=(C / D)-5;
15        B=(C+D)-5;
16      }
17      else
18      {
19      }
20      D=(A-B) % 5;
21      B=(A*B)+9;
22    }
23    else
24    {
25      if ((A > C) ^ (B > A) )
26      {
27        C=(D+C) % 2;
28        D=(A / D) % 3;
29      }
30      else
31      {
32        C=(C % B)+1;
33      }
34      A=(C-A)% 5;
35    }
36    std::cout << A, B, C, D;
37  }
```

курсы | Журнал компиляции | Отладка | Результаты поиска | Закрыть

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
```

C:\Users\ЧшырЄют√\Documents\defC\21-00.exe

```
4
--------------------------------
Process exited after 0.1726 seconds with return value 0
Для продолжения нажмите любую клавишу . . .
```

Пример выполнения (шаг 5)

5. Из реда… код в среду
программ… ветствии с синтаксисом языка
программ…

21-00.cpp

```cpp
#include "iostream"
#include <stdio.h>

int main()
{
    setlocale(LC_ALL, "Russian");
  int A = -5;
  int B = 19;
  int C = -40, D = 16;
  if (D > B)
  {
    if (D > A)
    {
      D=(C / D)-5;
      B=(C+D)-5;
    }
    else
    {
    }
    D=(A-B) % 5;
    B=(A*B)+9;
  }
  else
  {
    if ((A > C) ^ (B > A) )
    {
      C=(D+C) % 2;
      D=(A / D) % 3;
    }
    else
    {
      C=(C % B)+1;
    }
    A=(C-A)% 5;
  }
  std::cout << A, B, C, D;
}
```

…урсы | Журнал компиляции | Отладка | Результаты поиска | Закрыть

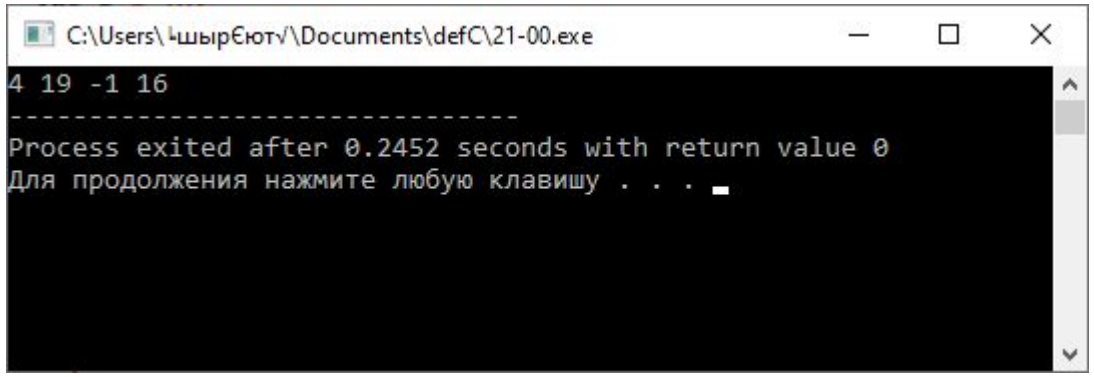
```
Compilation results...
--------
- Errors: 0
- Warnings: 0
```

C:\Users\…шырЄют√\Documents\defC\21-00.exe

```
4
--------------------------------
Process exited after 0.1726 seconds with return value 0
Для продолжения нажмите любую клавишу . . .
```

Пример выполнения (ша

5. Из редактора блок-схем скопировать исход
программирования. Дополнить программу в
программирования Си (C++).



21-00.cpp

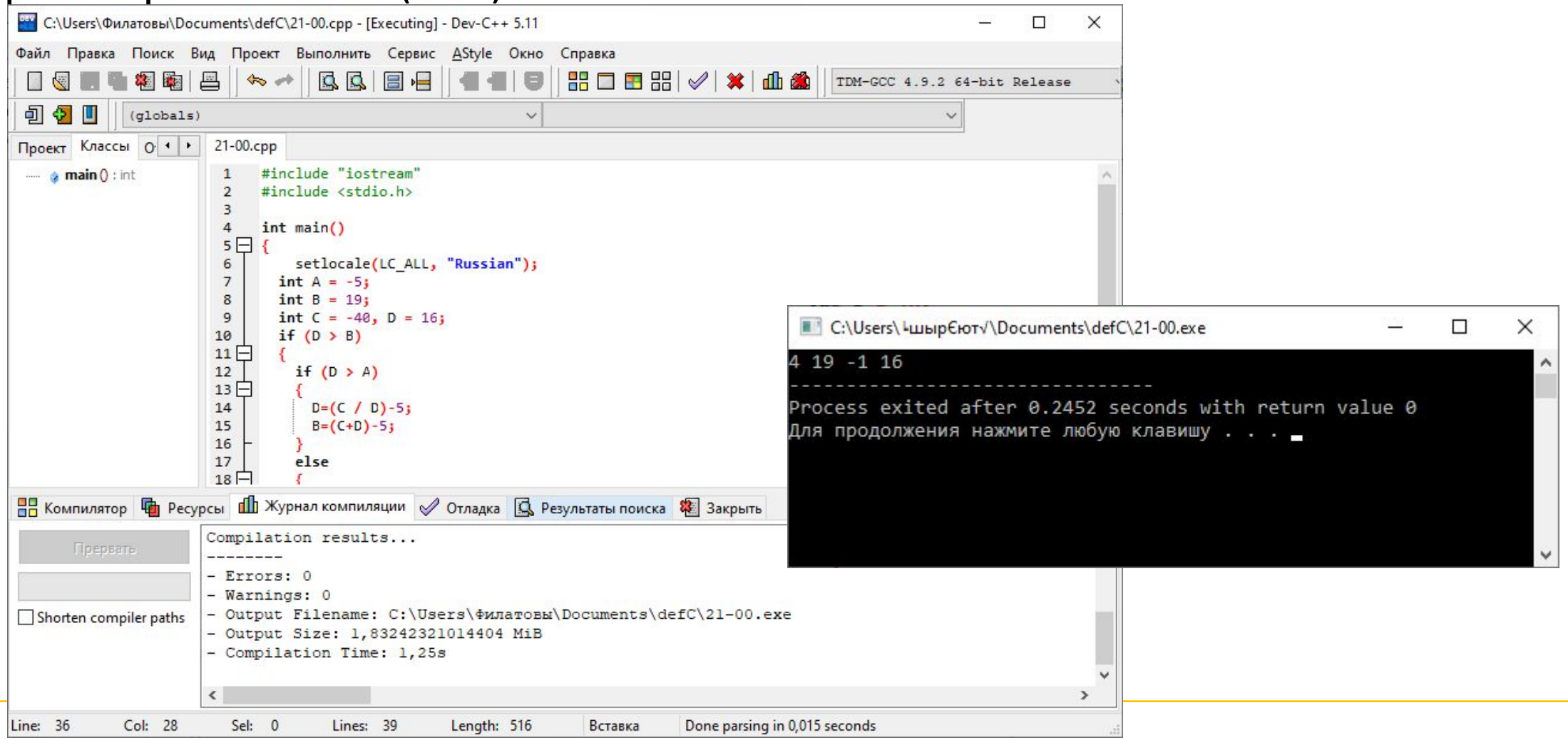
```
1  #include "iostream"
2  #include <stdio.h>
3
4  int main()
5  {
6      setlocale(LC_ALL, "Russian");
7    int A = -5;
8    int B = 19;
9    int C = -40, D = 16;
10   if (D > B)
11   {
12     if (D > A)
13     {
14       D=(C / D)-5;
15       B=(C+D)-5;
16     }
17     else
18     {
19     }
20     D=(A-B) % 5;
21     B=(A*B)+9;
22   }
23   else
24   {
25     if ((A > C) ^ (B > A) )
26     {
27       C=(D+C) % 2;
28       D=(A / D) % 3;
29     }
30     else
31     {
32       C=(C % B)+1;
33     }
34     A=(C-A)% 5;
35   }
36   std::cout << A <<" "<< B <<" "<< C << " "<< D;
37 }
```

урсы | Журнал компиляции | Отладка | Результаты поиска | Закрыть

```
Compilation results...
--------
- Errors: 0
```

Пример выполнения (шаг 6)

6. Из редактора блок-схем скопировать исходный код в среду программирования. Дополнить программу в соответствии с синтаксисом языка программирования Си (C++).



Пример выполнения (шаг 7)

7. Построить в отчете таблицу трассировки.

Таблица трассировки представляет собой таблицу, каждая ***строка*** которой ***соответствует*** выполнению отдельного оператора программы (или блока блок-схемы) или его части на очередном ***шаге*** выполнения алгоритма.

В таблице выделяют ***область протокола выполнения команды*** и ***область данных***.

Область команд				Область данных			
№ шага	№ блока	Оператор (операция)	Результат	Переменная			
				A	B	C	D

Пример выполнения (шаг 7)

7. Построить в отчете таблицу трассировки.

Таблица трассировки представляет собой таблицу, каждая строка которой соответствует выполнению отдельного оператора программы (или блока блок-схемы) или его части на очередном шаге выполнения алгоритма.

Область протокола выполнения команд состоит из

- ***№ шага*** - порядковый номер шага вычислительного процесса в соответствии с фактической очередностью выполнения команд алгоритма
- ***№ блока*** – идентификатор оператора в коде (например, номер строки программы или номер блока блок-схемы)
- ***Оператор*** – фактически выполняемая операция вычислительного процесса без учета текущего значения операндов
- ***Результат*** – процесс вычисления значения, которое будет определено как результат выполнения команды

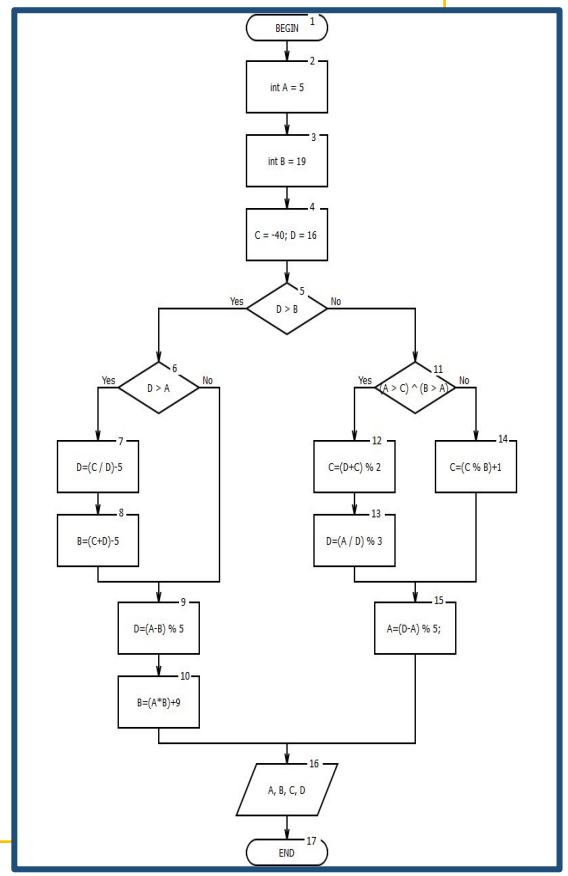
В ***области данных*** таблицы трассировки для ***каждой переменной*** из алгоритма ***определен столбец***, соответствующий содержимому ячейки памяти переменной

бласть данных.

Область команд				Область данных			
№ шага	№ блока	Оператор (операция)	Результат	Переменная			
				A	B	C	D

Пример выполнения (шаг 7)

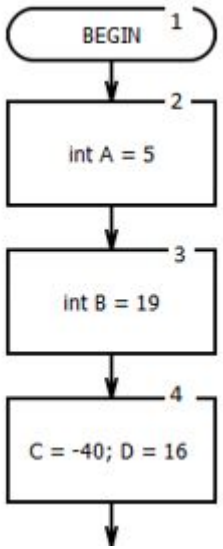
7. Построение таблицы трассировки

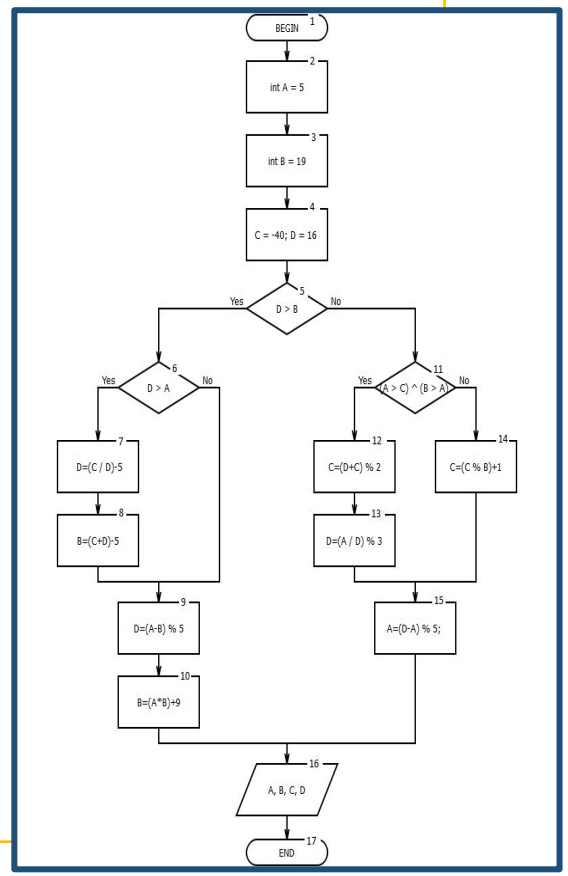


№ шага	№ блока	Оператор (операция)	Результат	Переменная			
				A	B	C	D
1	1	Начало					

Пример выполнения (шаг 7)

7. Построение таблицы трассировки

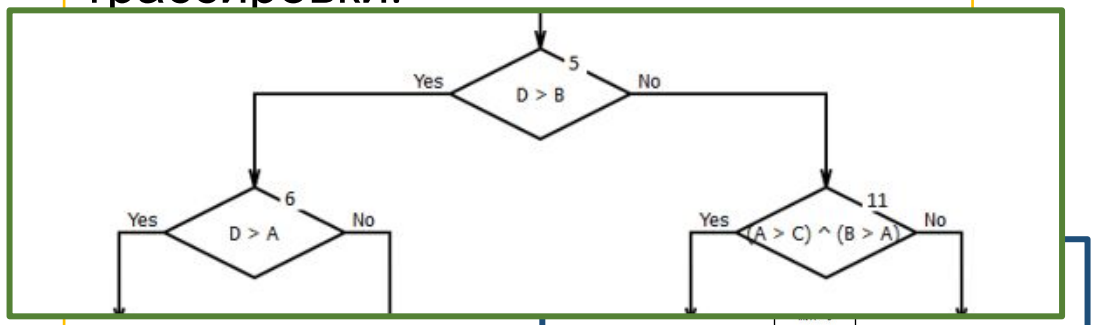


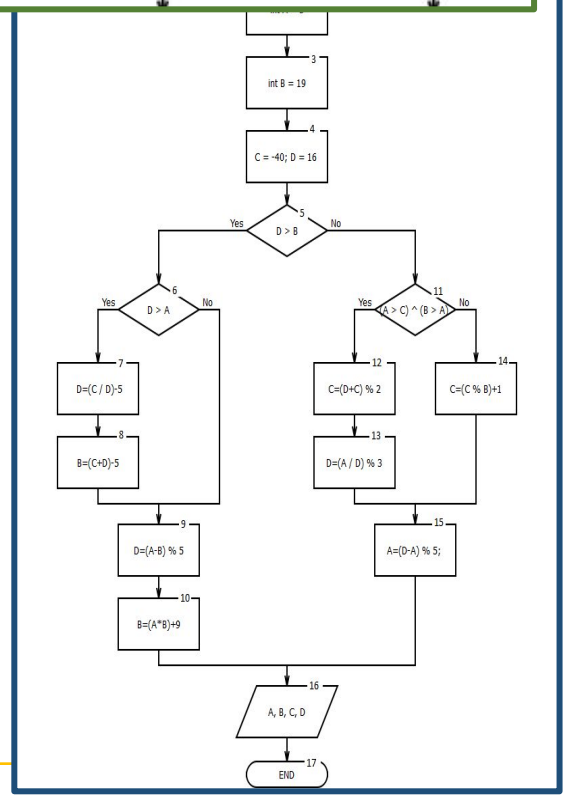


№ шага	№ блока	Оператор (операция)	Результат	Переменная			
				A	B	C	D
1	1	Начало					
2	2	Присваивание: A=5	A <- 5	5			
3	3	Присваивание: B=19	B <- 19		19		
4	4	Присваивание: C = -40 D= 16	C <- -40 D <- 16			-40	16

Пример выполнения (шаг 7)

7. Построение таблицы трассировки.

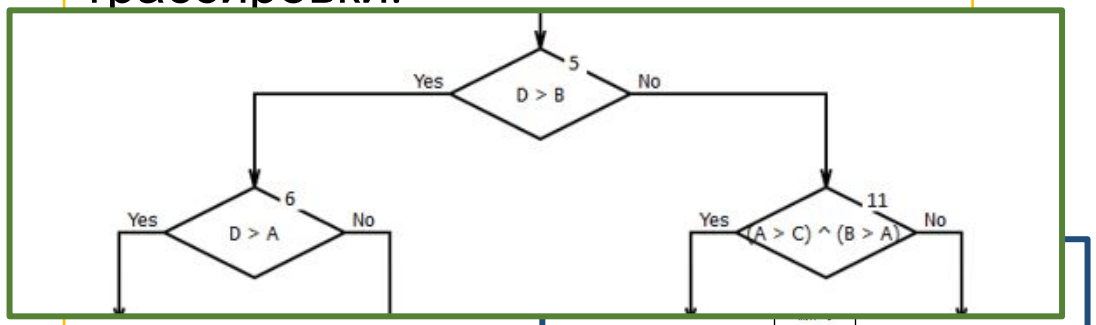


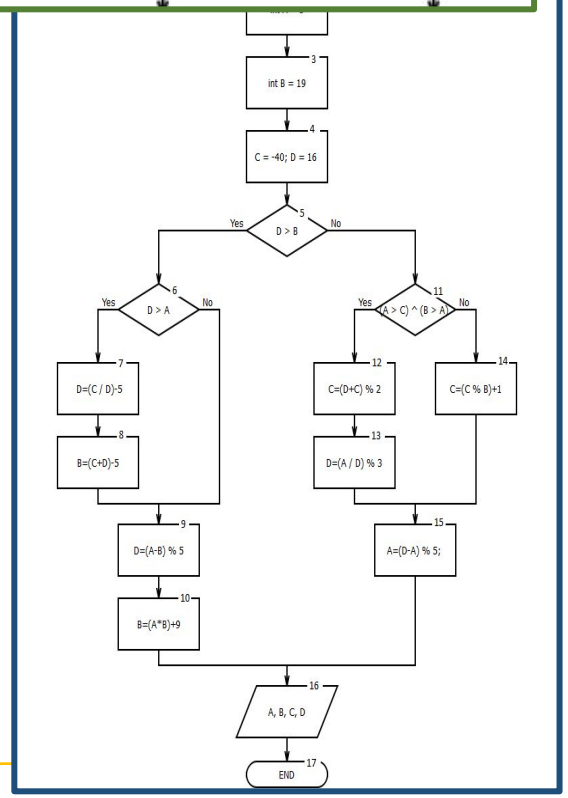


№ шага	№ блока	Оператор (операция)	Результат	Переменная			
				A	B	C	D
1	1	Начало					
2	2	Присваивание: A=5	A <- 5	5			
3	3	Присваивание: B=19	B <- 19		19		
4	4	Присваивание: C = -40 D= 16	C <- -40 D <- 16			-40	16
5	5	Проверка: D > B					

Пример выполнения (шаг 7)

7. Построение таблицы трассировки.

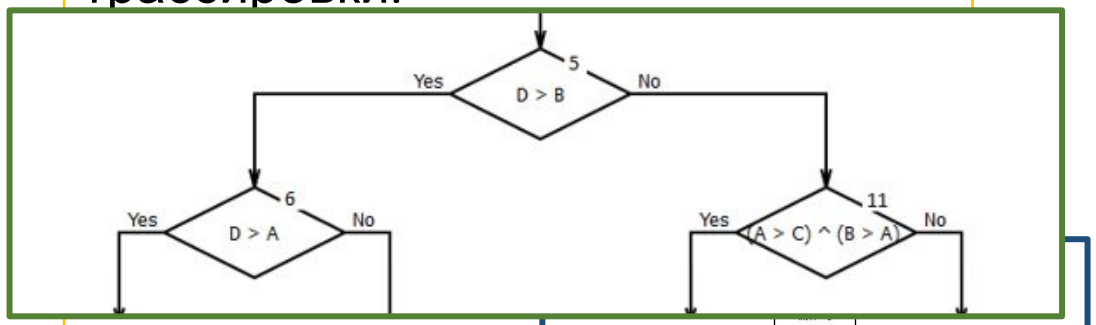


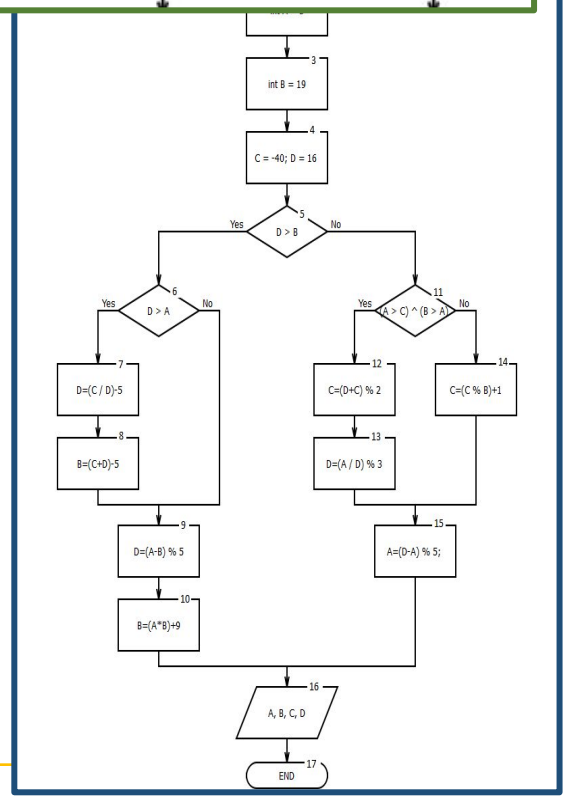


№ шага	№ блока	Оператор (операция)	Результат	Переменная			
				A	B	C	D
1	1	Начало					
2	2	Присваивание: A=5	A <- 5	5			
3	3	Присваивание: B=19	B <- 19		19		
4	4	Присваивание: C = -40 D= 16	C <- -40 D <- 16			-40	16
5	5	Проверка: D > B	16 > 19 --> НЕТ				

Пример выполнения (шаг 7)

7. Построение таблицы трассировки.

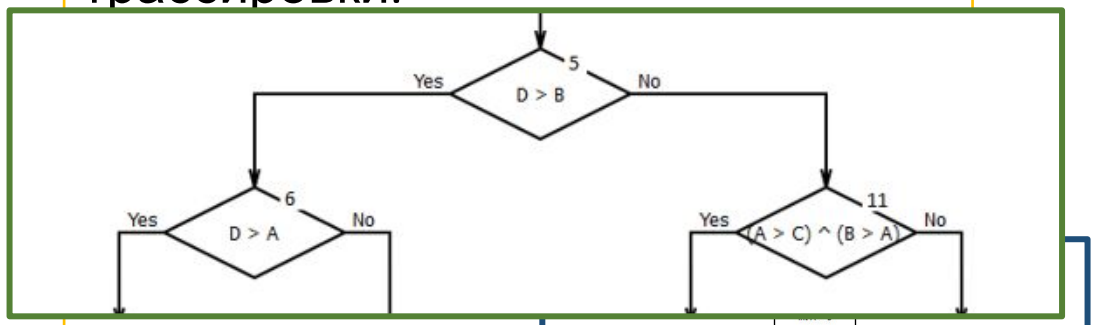


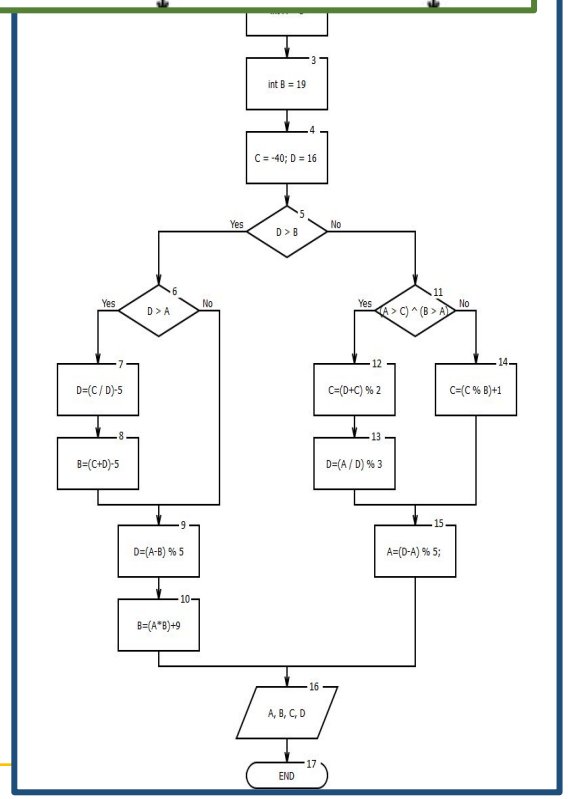


№ шага	№ блока	Оператор (операция)	Результат	Переменная			
				A	B	C	D
1	1	Начало					
2	2	Присваивание: A=5	A <- 5	5			
3	3	Присваивание: B=19	B <- 19		19		
4	4	Присваивание: C = -40 D= 16	C <- -40 D <- 16			-40	16
5	5	Проверка: D > B	16 > 19 --> НЕТ				
6	11	Проверка: (A > C) xor (B > A)	(5 > -40) xor (19 > 5) = Да xor да --> НЕТ				

Пример выполнения (шаг 7)

7. Построение таблицы трассировки.

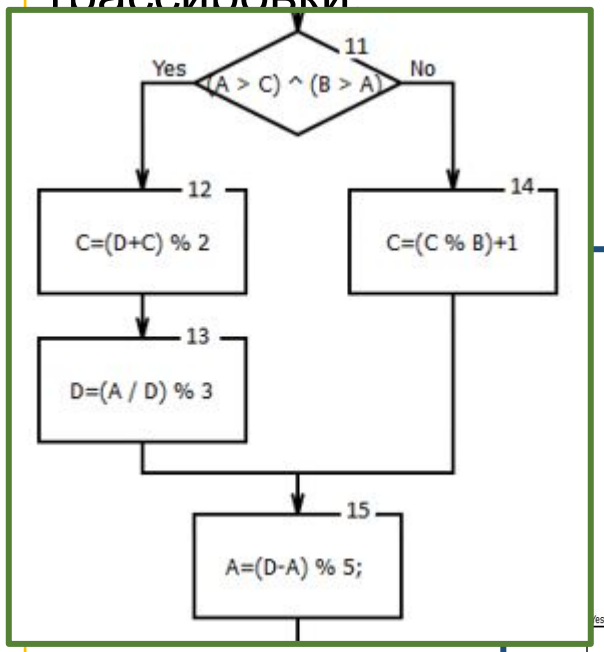


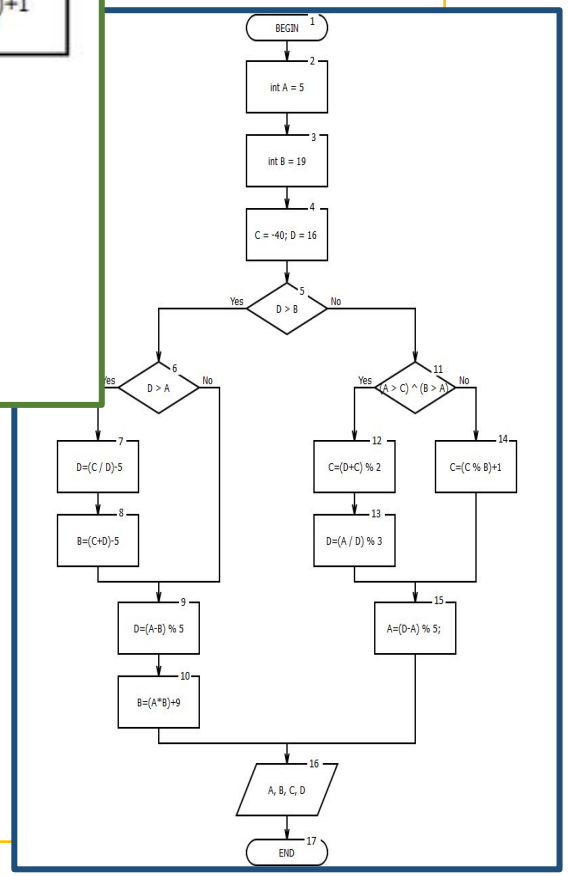


№ шага	№ блока	Оператор (операция)	Результат	Переменная			
				A	B	C	D
1	1	Начало					
2	2	Присваивание: A=5	A <- 5	5			
3	3	Присваивание: B=19	B <- 19		19		
4	4	Присваивание: C = -40 D= 16	C <- -40 D <- 16			-40	16
5	5	Проверка: D > B	16 > 19 --> НЕТ				
6	11	Проверка: (A > C) xor (B > A)	(5 > -40) xor (19 > 5) = Да xor да --> НЕТ				

Пример выполнения (шаг 7)

7. Построение таблицы трассировки

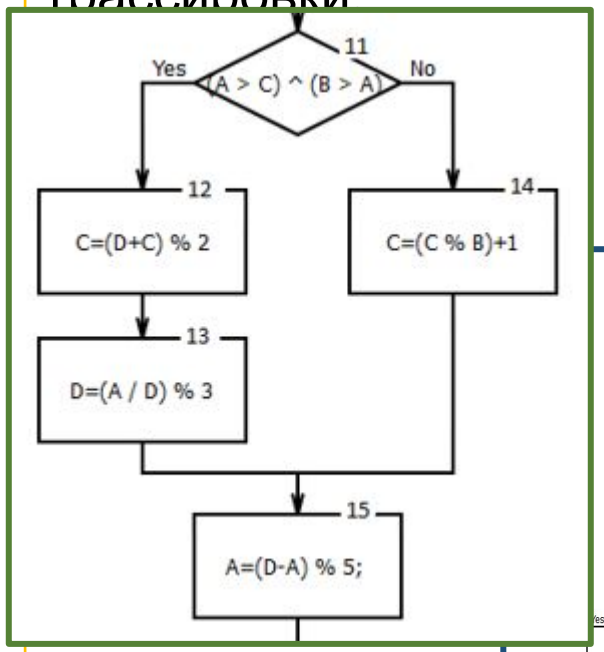


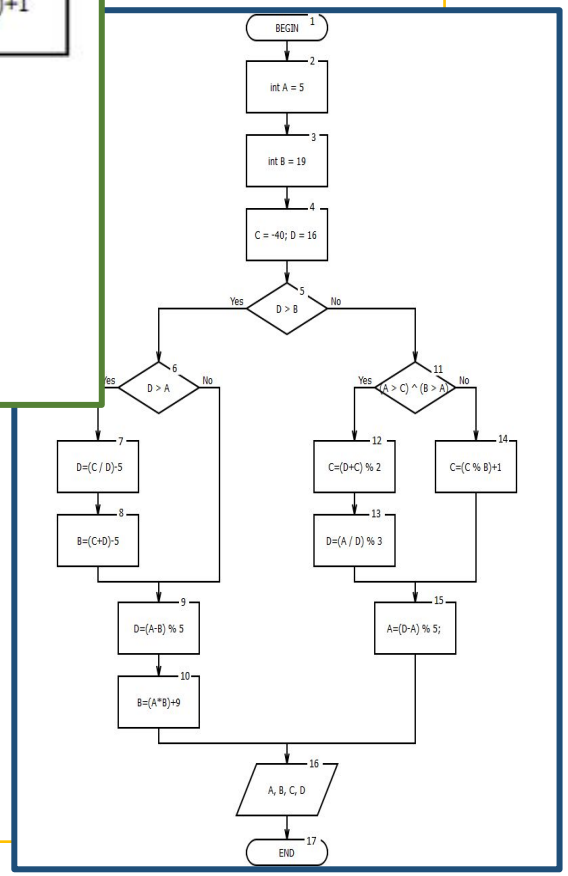


№ шага	№ блока	Оператор (операция)	Результат	Переменная			
				A	B	C	D
1	1	Начало					
2	2	Присваивание: A=5	A <- 5	5			
3	3	Присваивание: B=19	B <- 19		19		
4	4	Присваивание: C = -40 D= 16	C <- -40 D <- 16			-40	16
5	5	Проверка: D > B	16 > 19 --> НЕТ				
6	11	Проверка: (A > C) xor (B > A)	(5 > -40) xor (19 > 5) = Да xor да --> НЕТ				
7	14	Присваивание: C=(C mod B) +1	C <- (-40 mod 19) +1 = -2 + 1 = -1			-1	

Пример выполнения (шаг 7)

7. Построение таблицы трассировки

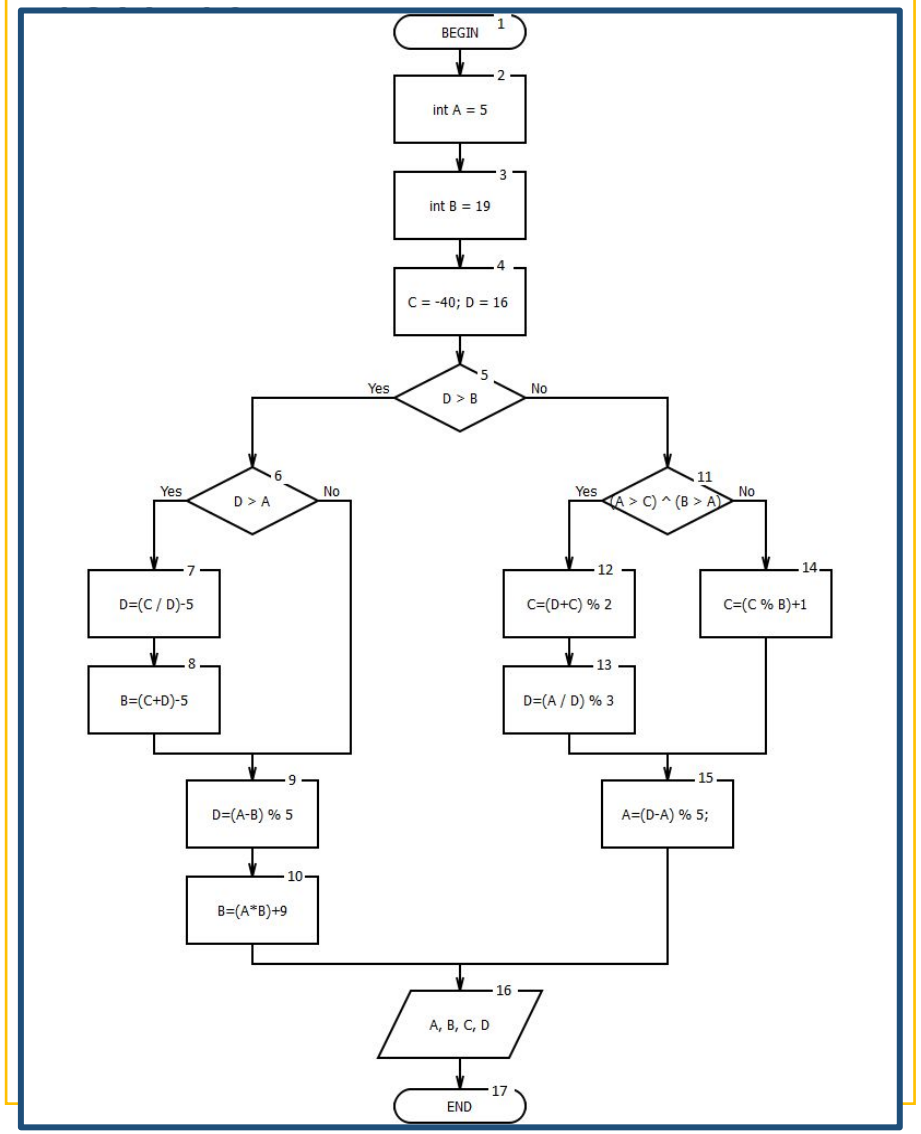




№ шага	№ блока	Оператор (операция)	Результат	Переменная			
				A	B	C	D
1	1	Начало					
2	2	Присваивание: A=5	A <- 5	5			
3	3	Присваивание: B=19	B <- 19		19		
4	4	Присваивание: C = -40 D= 16	C <- -40 D <- 16			-40	16
5	5	Проверка: D > B	16 > 19 --> НЕТ				
6	11	Проверка: (A > C) xor (B > A)	(5 > -40) xor (19 > 5) = Да xor да --> НЕТ				
7	14	Присваивание: C=(C mod B) +1	C <- (-40 mod 19) +1 = -2 + 1 = -1			-1	
8	15	Присваивание: A = (D – A) mod 5	A <- (16 -5) mod 5 = 11 mod 5 = 1	1			

Пример выполнения (шаг 7)

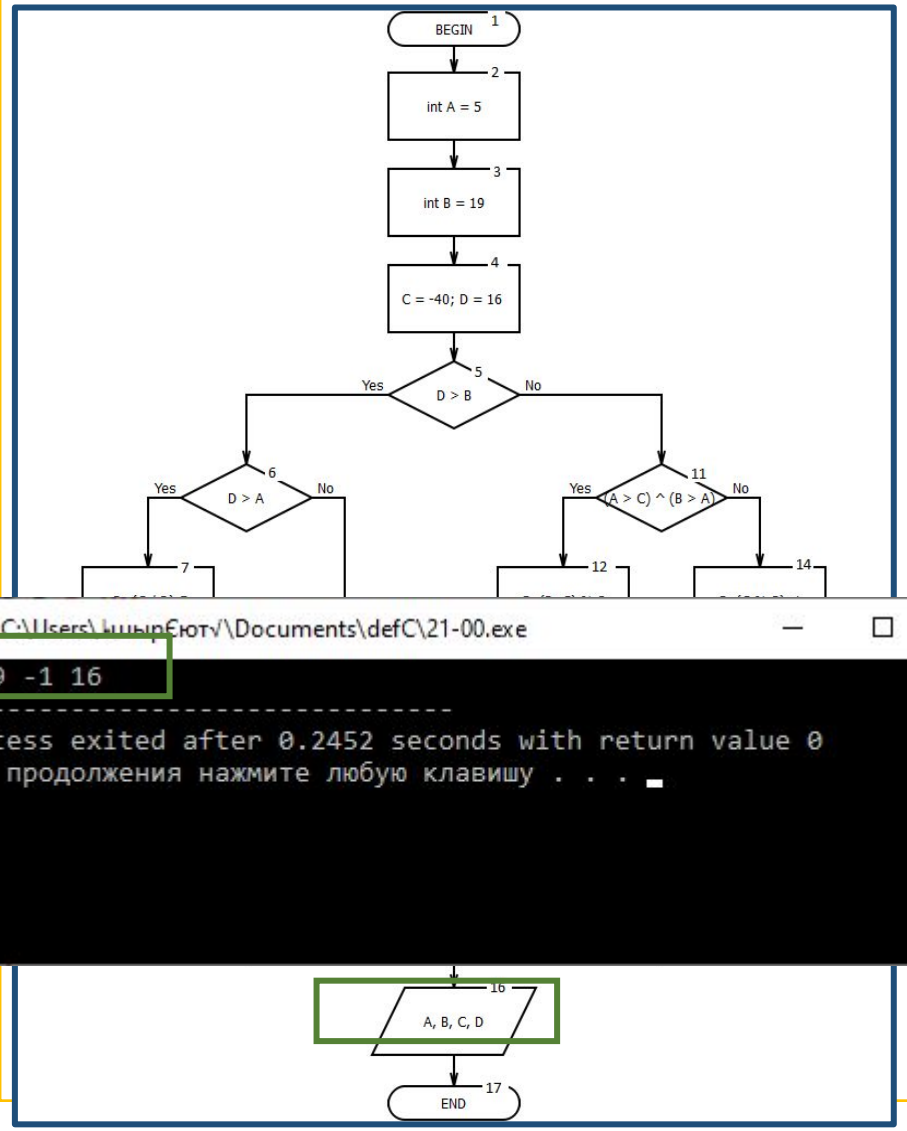
7. Построение таблицы

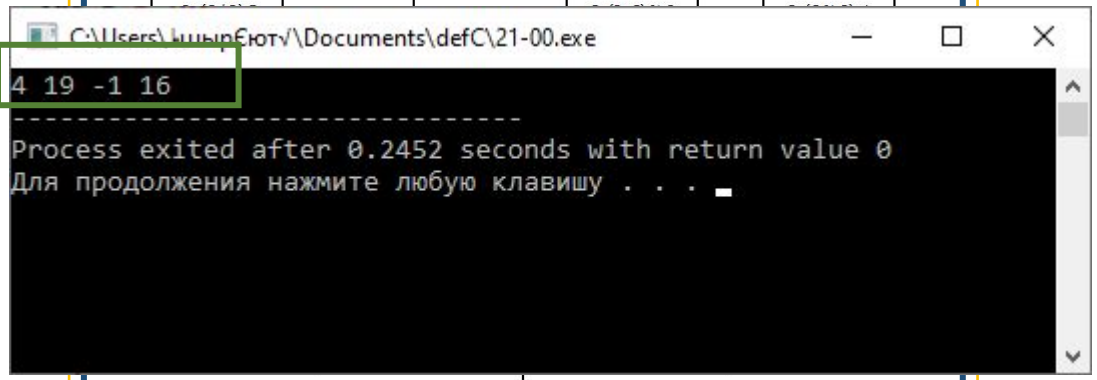


№ шага	№ блока	Оператор (операция)	Результат	Переменная			
				A	B	C	D
1	1	Начало					
2	2	Присваивание: A=5	A <- 5	5			
3	3	Присваивание: B=19	B <- 19		19		
4	4	Присваивание: C = -40 D= 16	C <- -40 D <- 16			-40	16
5	5	Проверка: D > B	16 > 19 --> НЕТ				
6	11	Проверка: (A > C) xor (B > A)	(5 > -40) xor (19 > 5) = Да xor да --> НЕТ				
7	14	Присваивание: C=(C mod B) +1	C <- (-40 mod 19) +1 = -2 + 1 = -1			-1	
8	15	Присваивание: A = (D – A) mod 5	A <- (16 -5) mod 5 = 11 mod 5 = 1	1			
9	16	Печать: A, B, C, D	Печать: 1 19 -1 16				
10	17	Конец					

Пример выполнения (шаг 8)

8. Сравнение результатов





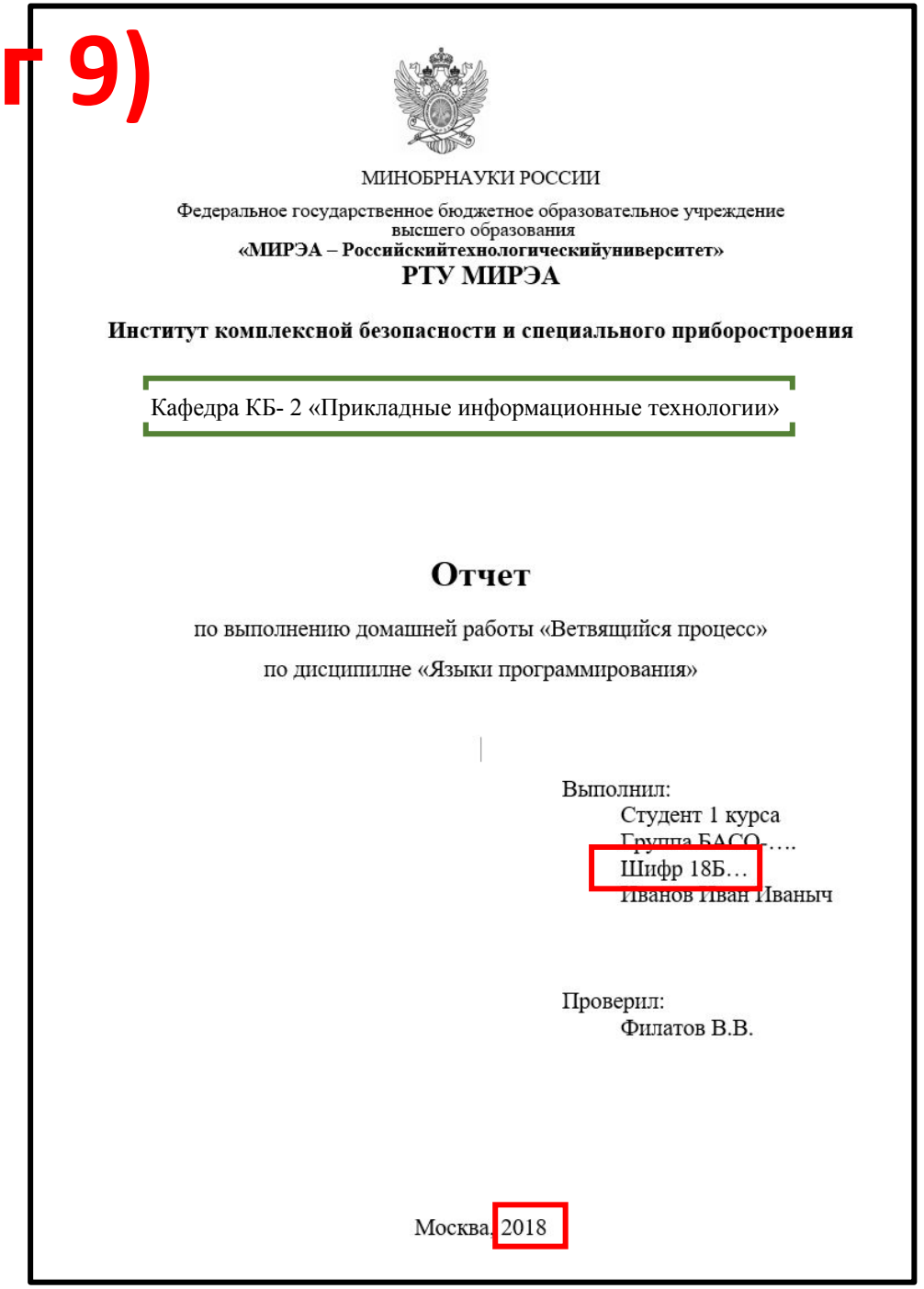
№ шага	№ блока	Оператор (операция)	Результат	Переменная			
				A	B	C	D
1	1	Начало					
2	2	Присваивание: A=5	A <- 5	5			
3	3	Присваивание: B=19	B <- 19		19		
4	4	Присваивание: C = -40 D= 16	C <- -40 D <- 16			-40	16
5	5	Проверка: D > B	16 > 19 --> НЕТ				
6	11	Проверка: (A > C) xor (B > A)	(5 > -40) xor (19 > 5) = Да xor да --> НЕТ				
7	14	Присваивание: C=(C mod B) +1	C <- (-40 mod 19) +1 = -2 + 1 = -1			-1	
8	15	Присваивание: A = (D – A) mod 5	A <- (16 -5) mod 5 = 11 mod 5 = 1	1			
9	16	Печать: A, B, C, D	Печать: 1 19 -1 16				
10	17	Конец					

Пример выполнения (шаг 9)

9. Печать отчета

БУМАЖНЫЙ отчет:

- Титульник отчета
- Задание с номером варианта
- Структурированное представление задания
- Блок-схема с номерами блоков
- Скриншот исходника в среде программирования с результатом работы программы
- Таблица трассировки
- Выводы
- Источники (литература)



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российскийтехнологическийуниверситет»
РТУ МИРЭА

Институт комплексной безопасности и специального приборостроения

Кафедра КБ- 2 «Прикладные информационные технологии»

Отчет

по выполнению домашней работы «Ветвящийся процесс»

по дисциплине «Языки программирования»

Выполнил:
Студент 1 курса
Группа БАСО-…
Шифр 18Б…
Иванов Иван Иваныч

Проверил:
Филатов В.В.

Москва, 2018