

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
дисциплины «Алгоритмизация»

Выполнил:
Гайчук Дарья Дмитриевна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика
и вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А.-доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тип 16 № 7340

Алгоритм вычисления значения функции $F(n)$, где n — натуральное число, задан следующими соотношениями:
 $F(1) = 1$
 $F(n) = F(n-1) + 2^{n-1}$, если $n > 1$.
Чему равно значение функции $F(12)$?
В ответе запишите только натуральное число.

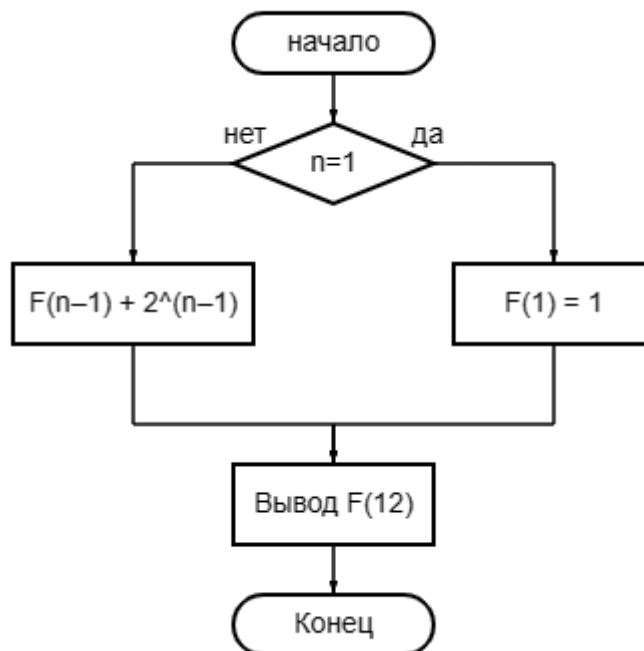


Рисунок 1. Блок схема алгоритма №1

```
7340.cpp 7340 (Глобальная область) calculateF(int n)
1  #include <stdio.h>
2  #include <cmath>
3
4  int calculateF(int n) {
5      if (n == 1) {
6          return 1;
7      }
8      else {
9          return calculateF(n - 1) + (pow(2, (n - 1)));
10     }
11 }
12
13 int main() {
14     int n = 12;
15     int result = calculateF(n);
16
17     printf("F(%d) = %d\n", n, result);
18
19     return 0;
20 }
```

Рисунок 2. Код программы №1


```
37372.cpp
37372 (Глобальная область) main()
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     int n = 10000;
6     int maxDifference = 0;
7     int countPairs = 0;
8
9     int* sequence = (int*)malloc(n * sizeof(int));
10
11     for (int i = 0; i < n; i++) {
12         sequence[i] = rand() % 10000 + 1;
13     }
14
15     for (int i = 0; i < n; i++) {
16         for (int j = i + 1; j < n; j++) {
17             int difference = abs(sequence[i] - sequence[j]);
18
19             if ((difference % 45 == 0) && (sequence[i] % 18 == 0 || sequence[j] % 18 == 0)) {
20                 countPairs++;
21             }
22
23             if (difference > maxDifference) {
24                 maxDifference = difference;
25             }
26         }
27     }
28     printf("Number of pairs with difference a multiple of 45 and at least one multiple of 18: %d\n", countPairs);
29     printf("Maximum difference among such pairs: %d\n", maxDifference);
30     return 0;
31 }
```

Рисунок 5. Код программы №2

```
Консоль отладки Microsoft
Number of pairs with difference a multiple of 45 and at least one multiple of 18: 94830
Maximum difference among such pairs: 9945

C:\Users\doris\source\repos\37372\x64\Debug\37372.exe (процесс 31820) завершил работу с кодом
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 6. Результат выполнения программы №2

Тип 18 № 27683

Квадрат разлинован на $N \times N$ клеток ($1 < N < 17$). Исполнитель Робот может перемещаться по клеткам, выполняя за одно перемещение одну из двух команд: вправо или вниз. По команде вправо Робот перемещается в соседнюю правую клетку, по команде вниз — в соседнюю нижнюю. При попытке выхода за границу квадрата Робот разрушается. Перед каждым запуском Робота в каждой клетке квадрата лежит монета достоинством от 1 до 100. Посетив клетку, Робот забирает монету с собой; это также относится к начальной и конечной клетке маршрута Робота.

Задание 18

Откройте файл. Определите максимальную и минимальную денежную сумму, которую может собрать Робот, пройдя из левой верхней клетки в правую нижнюю. В ответ запишите два числа друг за другом без разделительных знаков — сначала максимальную сумму, затем минимальную.

Исходные данные представляют собой электронную таблицу размером $N \times N$, каждая ячейка которой соответствует клетке квадрата.

Пример входных данных:

1	8	8	4
10	1	1	3
1	3	12	2
2	3	5	6

Для указанных входных данных ответом должна быть пара чисел 41 и 22.

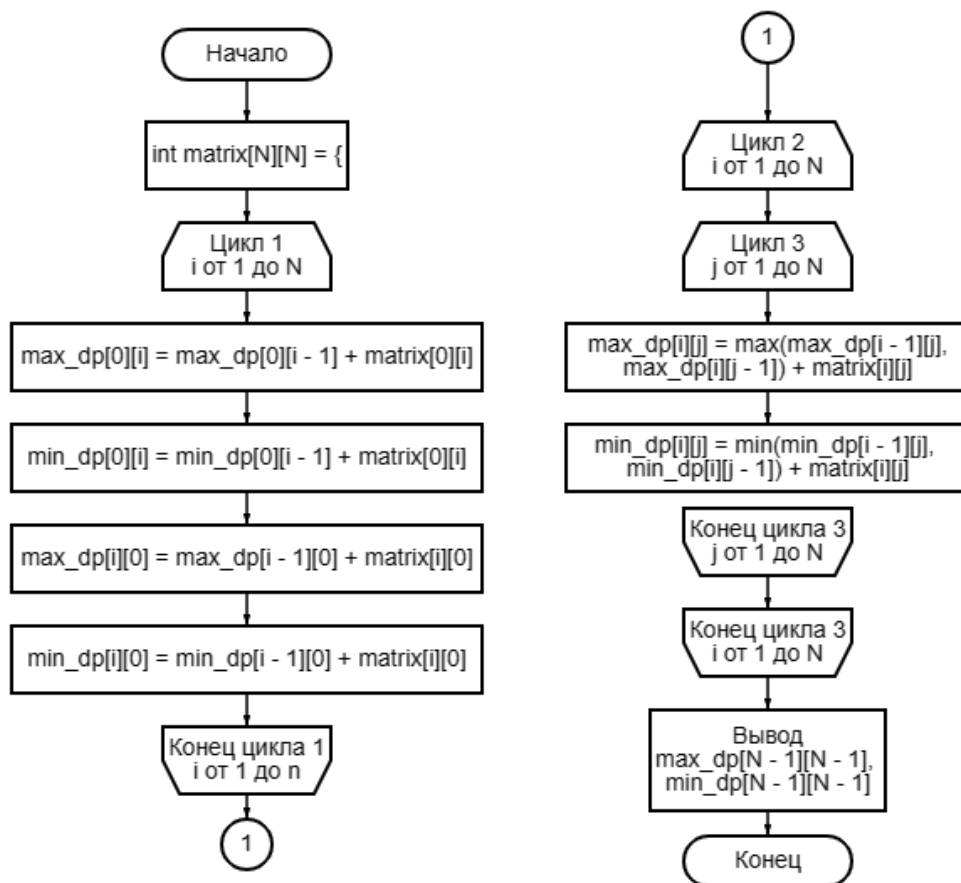


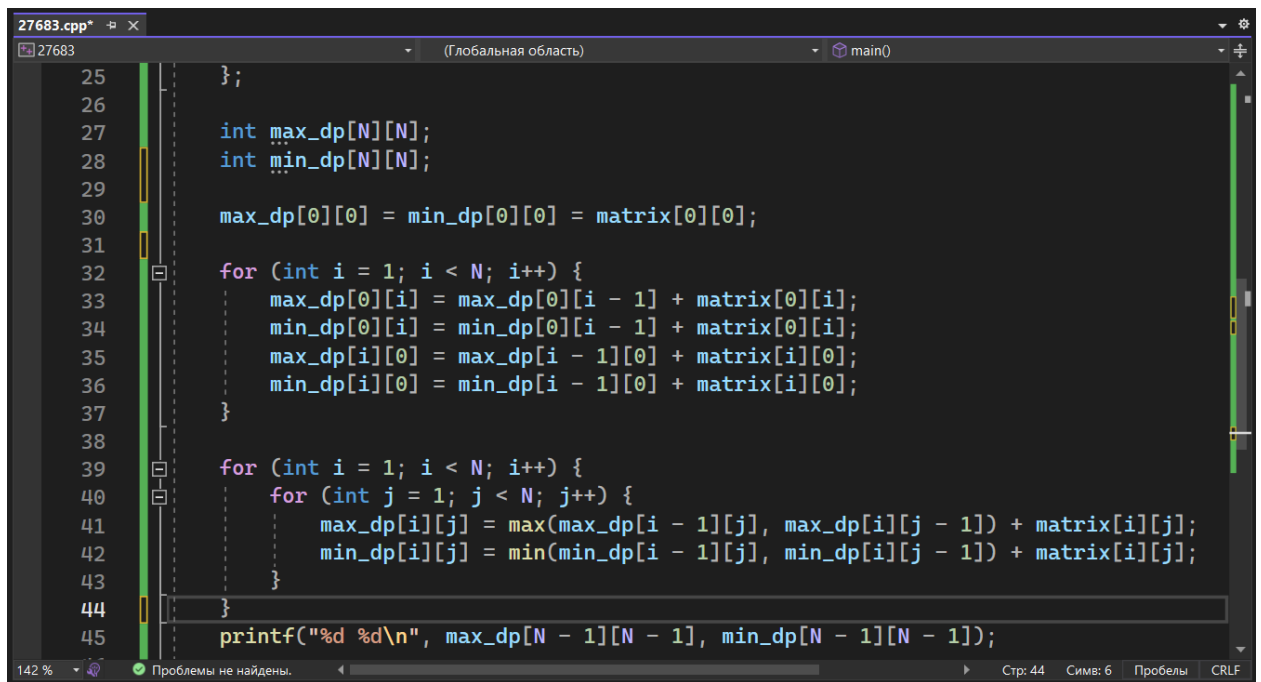
Рисунок 7. Блок схема алгоритма №3

```

27683.cpp
27683
(Глобальная область)
main()

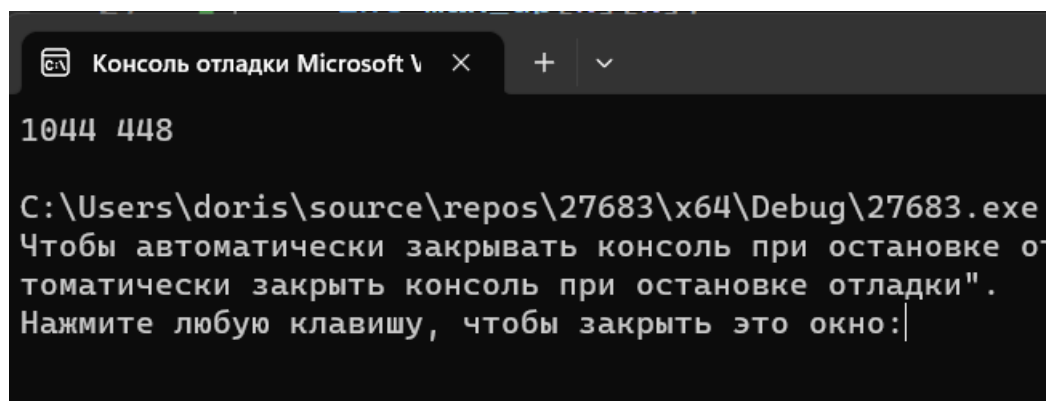
5  int max(int a, int b) {
6      return (a > b) ? a : b;
7  }
8
9  int min(int a, int b) {
10     return (a < b) ? a : b;
11 }
12
13 int main() {
14     int matrix[N][N] = {
15         {2, 79, 5, 75, 7, 98, 53, 92, 72, 39},
16         {48, 31, 23, 15, 98, 87, 3, 3, 47, 26},
17         {6, 12, 59, 2, 33, 66, 42, 69, 75, 1},
18         {48, 88, 45, 64, 17, 43, 83, 11, 1, 71},
19         {66, 1, 27, 14, 2, 8, 41, 11, 15, 28},
20         {35, 1, 64, 73, 26, 34, 10, 12, 31, 91},
21         {8, 37, 48, 2, 50, 41, 65, 33, 86, 4},
22         {36, 64, 5, 99, 94, 70, 33, 50, 98, 16},
23         {28, 1, 69, 16, 25, 63, 19, 21, 84, 56},
24         {79, 39, 15, 9, 77, 6, 12, 72, 32, 18}
25     };
  
```

Рисунок 8. Задание пути робота



```
25     };
26
27     int max_dp[N][N];
28     int min_dp[N][N];
29
30     max_dp[0][0] = min_dp[0][0] = matrix[0][0];
31
32     for (int i = 1; i < N; i++) {
33         max_dp[0][i] = max_dp[0][i - 1] + matrix[0][i];
34         min_dp[0][i] = min_dp[0][i - 1] + matrix[0][i];
35         max_dp[i][0] = max_dp[i - 1][0] + matrix[i][0];
36         min_dp[i][0] = min_dp[i - 1][0] + matrix[i][0];
37     }
38
39     for (int i = 1; i < N; i++) {
40         for (int j = 1; j < N; j++) {
41             max_dp[i][j] = max(max_dp[i - 1][j], max_dp[i][j - 1]) + matrix[i][j];
42             min_dp[i][j] = min(min_dp[i - 1][j], min_dp[i][j - 1]) + matrix[i][j];
43         }
44     }
45     printf("%d %d\n", max_dp[N - 1][N - 1], min_dp[N - 1][N - 1]);
```

Рисунок 9. Код программы №3



```
Консоль отладки Microsoft V x + v
1044 448
C:\Users\doris\source\repos\27683\x64\Debug\27683.exe
Чтобы автоматически закрывать консоль при остановке отладки, нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 10. Результат выполнения программы №3