

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчёт по лабораторной работе №2

по дисциплине

«Программирование»

Разработал студент гр. ИВТб-1301-05-00 _____ /Черкасов А. А./
(подпись)

Зав. кафедры ЭВМ _____ /Долженкова М. Л./
(подпись)

Киров

2024

Цель

Цель работы: изучение и применение основ программирования на языках Pascal и C через выполнение практических заданий, направленных на развитие навыков алгоритмического мышления, освоение синтаксиса языков, а также на решение задач, связанных с обработкой данных и реализацией алгоритмов.

Задания

1. Сумма двух предыдущих.

Задан числовой одномерный массив-кольцо, насчитывающий N элементов. Вместо каждого элемента с нулевым значением поставить сумму двух предыдущих элементов массива. Гарантируете, что количество элементов в массиве не менее трех.

Формат ввода.

В первой строке вводится натуральное число N - количество элементов в массиве. Во второй строке элементы массива через пробел.

Формат вывода.

В строке выводится новый массив элементов, разделяемых пробелом

Ввод	Вывод
3	3 1 2
0 1 2	

2. Убывающая последовательность.

Дан одномерный массив из N элементов. Определить, образуют ли элементы массива расположенные перед первым отрицательным числом, убывающую последовательность.

Формат ввода.

В первой строке вводится число N количество элементов в исходном массиве.
Во второй строке через пробел задаются элементы массива.

Формат вывода.

Вывести Yes - если найдена убывающая последовательность и No - в любом другом случае.

Ввод	Вывод
3	Yes
19 10 -4	
8	No
15 6 23 -19 -25 -4 -25 29	

3. Единицы.

Дан массив из N чисел, содержащий только нули и единицы. Найти номер элемента с которого начинается самая длинная последовательность единиц и количество элементов этой последовательности. Если таких последовательностей несколько, вывести номер последней из них. Если единицы в исходном массиве отсутствуют - вывести дважды 0.

Формат ввода.

В первой строке вводится число N количество элементов в исходном массиве.
Во второй строке через пробел задаются элементы массива.

Формат вывода.

В единственной строке через пробел записываются два целых числа - номер элемента (начиная с нулевого) с которого начинается последовательность единиц и длинна максимальной последовательности. Или 0 0 - если единиц в исходном массиве нет.

Ввод	Вывод
16	11 2
0 1 1 0 1 0 0 1 0 1 0 1 1 0 0 1	

4. Меньше правого соседа.

Дан массив из N целых чисел. Сформировать новый массив, состоящий из элементов исходного массив, значения которых меньше их правого соседа.

Формат ввода.

В первой строке задается число N - количество элементов исходного массива
Во второй строке через пробел задаются элементы массива.

Формат вывода.

В первой строке выводится длина нового массива Во второй строке элементы нового массива через пробел Если элементов удовлетворяющих условию нет, то в первой строке выводится NO, а вторая строка отсутствует.

Ввод	Вывод
15	5
9 -1 -3 -1 7 1 3 3 3 7 4 1 5 4 -10	-3 -1 1 3 1
11	6
-8 8 -1 10 0 1 6 -8 -5 5 -6	-8 -1 0 1 -8 -5

5. Меньше K.

В массиве из N целых чисел выбрать максимальное количество подряд идущих элементов, сумма которых не превышала бы целого числа K.

Формат ввода.

В первой строке вводится число N количество элементов в исходном массиве и число K. Во второй строке через пробел задаются элементы массива.

Формат вывода.

В единственной строке указать число соответствующее максимальному коли-

честву элементов, удовлетворяющих условию.

Ввод	Вывод
2 18 14 -24	2
16 14 7 8 -4 24 -25 -7 -8 12 22 -11 25 -18 15 25 5 15	10

6. Расширение массива.

В заданном одномерном массиве, состоящем из N целых чисел подсчитать количество элементов, делящихся нацело на 3, и целую часть (округление по правилам арифметики) среднего арифметического элементов с чётными значениями. Поставить полученные величины на первое и последнее места в исходном массиве (увеличить массив на 2 элемента).

Формат ввода.

В первой строке задается натуральное число $N \leq 200$ - количество элементов в исходном массиве. Во второй строке элементы массива, разделяемые пробелом.

Формат вывода.

В единственной строке вывести новый массив, элементы которого разделены пробелом.

Ввод	Вывод
4	1 5 17 -18 2 -8
5 17 -18 2	

7. Матрица средних.

В прямоугольной матрице A , имеющей N строк и M столбцов найти наименьшее значение среди средних значений для каждой строки матрицы.

Формат ввода.

В первой строке вводятся числа N и M - количество строк и столбцов матрицы

В следующих последовательных N строках вводятся через пробел элементы строк матрицы, начиная с первой строки.

Формат вывода.

Единственное число соответствующее среднему значению, округленному до двух знаков после запятой.

Ввод	Вывод
3 4	0.50
10 5 -4 2	
1 8 6 -13	
5 2 7 9	

8. Между нулями.

Дан массив из N целых чисел, содержащий по крайней мере два нуля. Вывести сумму чисел из данного массива, расположенных между двумя последними нулями.

Формат ввода.

В первой строке число N - количество элементов массива Во второй строке целые элементы массива через пробел.

Формат вывода.

Вывести единственное целое число, соответствующее искомой сумме.

Ввод	Вывод
9	0
19 10 16 10 1 20 10 0 0	
13	18
1 11 13 0 18 20 0 18 21 0 18 0 1	
13	141
0 14 7 11 19 1 21 14 6 4 21 23 0	

9. Рыба-пила.

Дан массив из N вещественных чисел ($N < 100$). Будем называть массив пилообразным, если каждый его внутренний элемент либо больше, либо меньше обоих его соседей. Например, массив 1 5 2 4 3 7 5 - пилообразный.

Формат ввода.

В первой строке вводится число N - количество элементов в массиве. Во второй строке вводятся через пробел элементы массива.

Формат вывода.

Вывести Yes - если массив пилообразный и No в противном случае.

Ввод	Вывод
5	Yes
1 0 1 0 1	

10. Нарушение закономерности.

Дан массив из N вещественных чисел. Если данный массив образует неубывающую последовательность, то вывести 0. В противном случае вывести номер первого числа (нумерация начинается с нуля) нарушающего закономерность.

Формат ввода.

В первой строке вводится число N - количество элементов в массиве. Во второй строке через пробел вводятся вещественные числа образующие массив.

Формат вывода.

Вывести 0 или номер элемента нарушающего закономерность.

Ввод	Вывод
15	2
-4.23 5.14 -1.94 6.39 -5.90 -7.38 -8.42	
-0.60 6.55 3.65 -5.96 8.27 1.8 -4.5 1.58	

Задание 1.

Схема алгоритма Задания 1 представлена на Рисунке 1.

Исходный код представлен в Приложении А1.

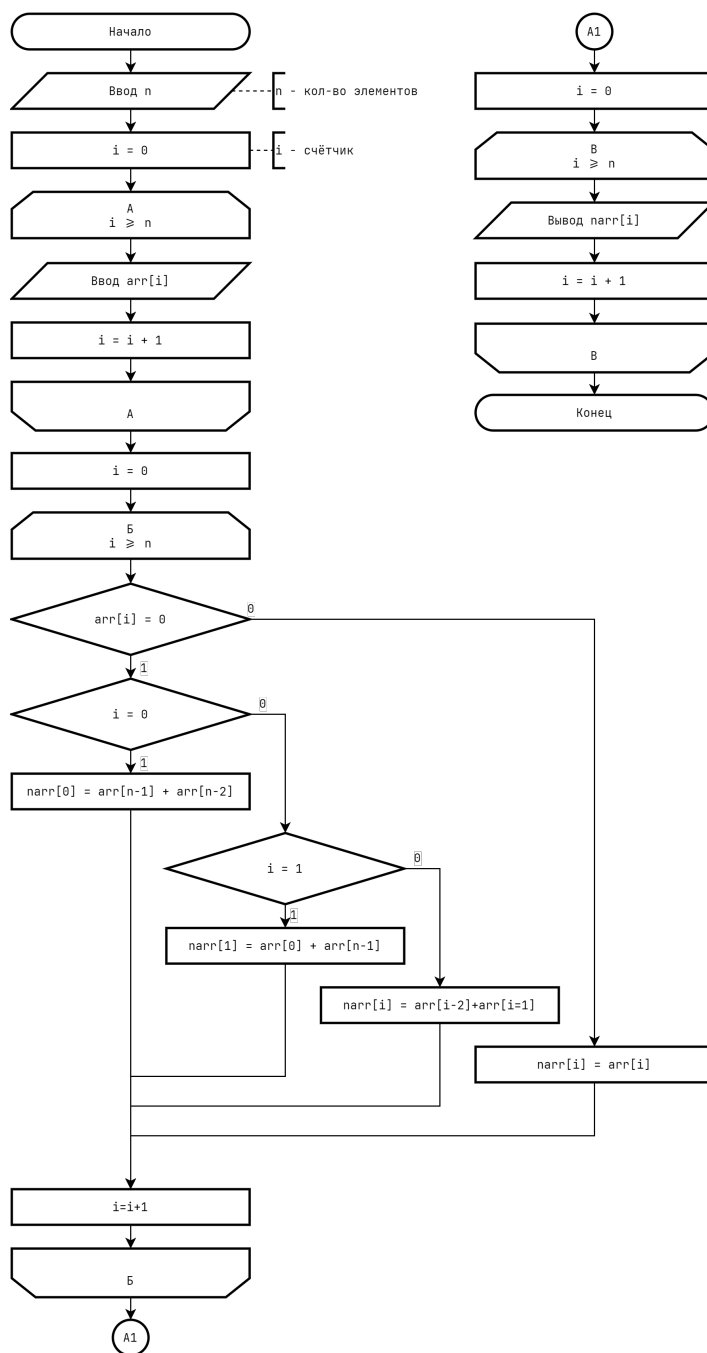


Рисунок 1 - Схема алгоритма Задания 1.

Задание 2.

Схема алгоритма Задания 2 представлена на Рисунке 2.

Исходный код представлен в Приложении А2.

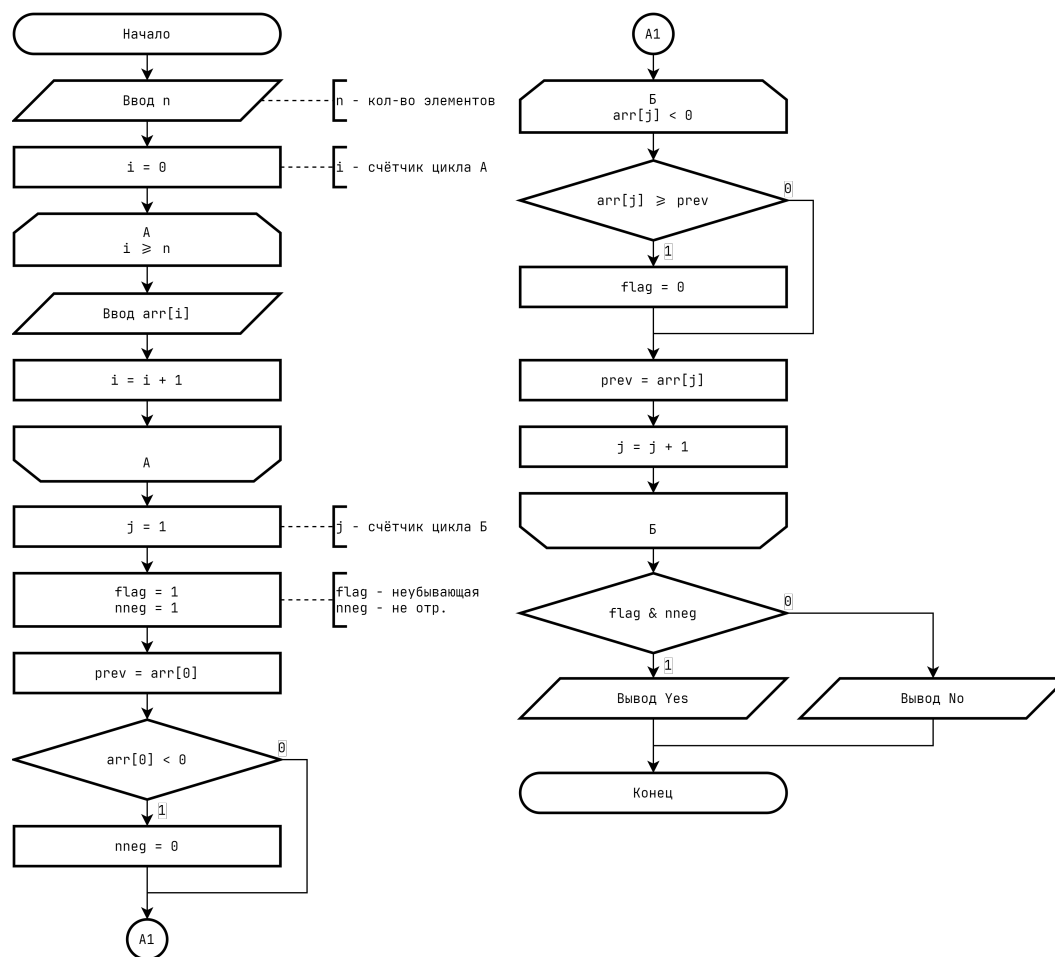


Рисунок 2 - Схема алгоритма Задания 2.

Задание 3.

Схема алгоритма Задания 3 представлена на Рисунке 3.

Исходный код представлен в Приложении А3.

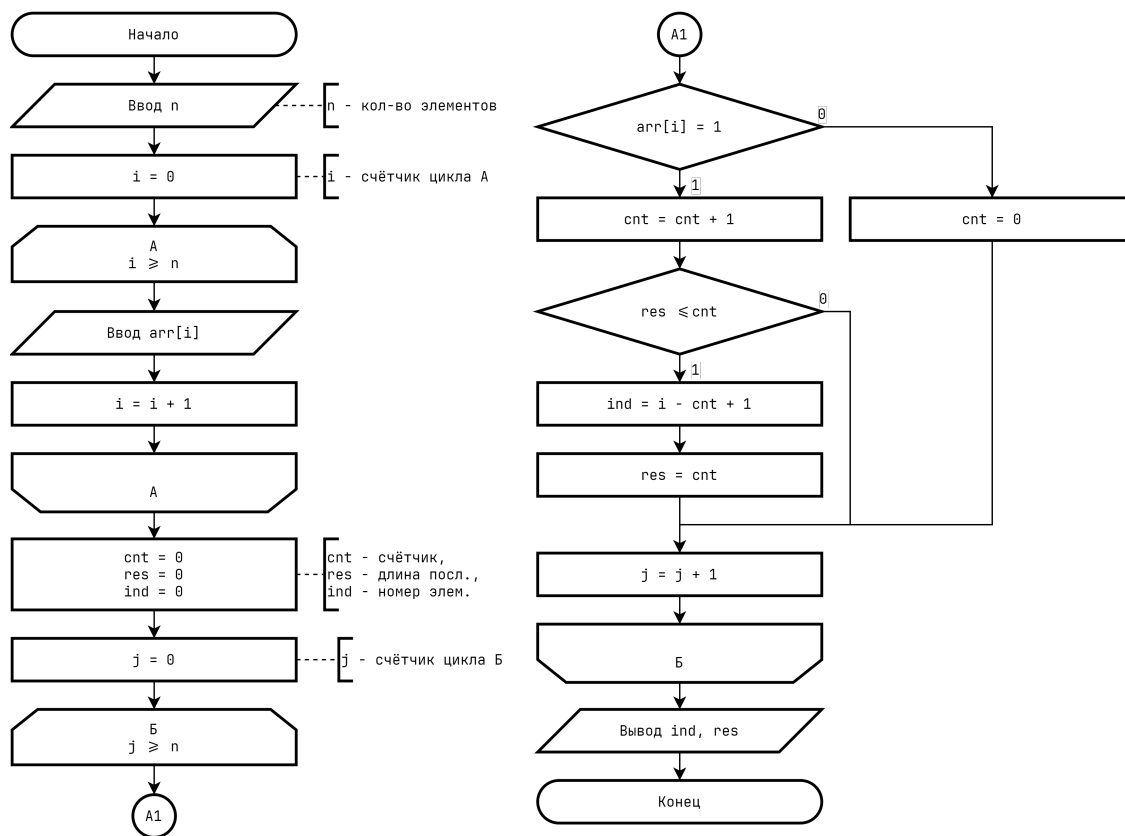


Рисунок 3 - Схема алгоритма Задания 3.

Задание 4.

Схема алгоритма Задания 4 представлена на Рисунке 4.

Исходный код представлен в Приложении А4.

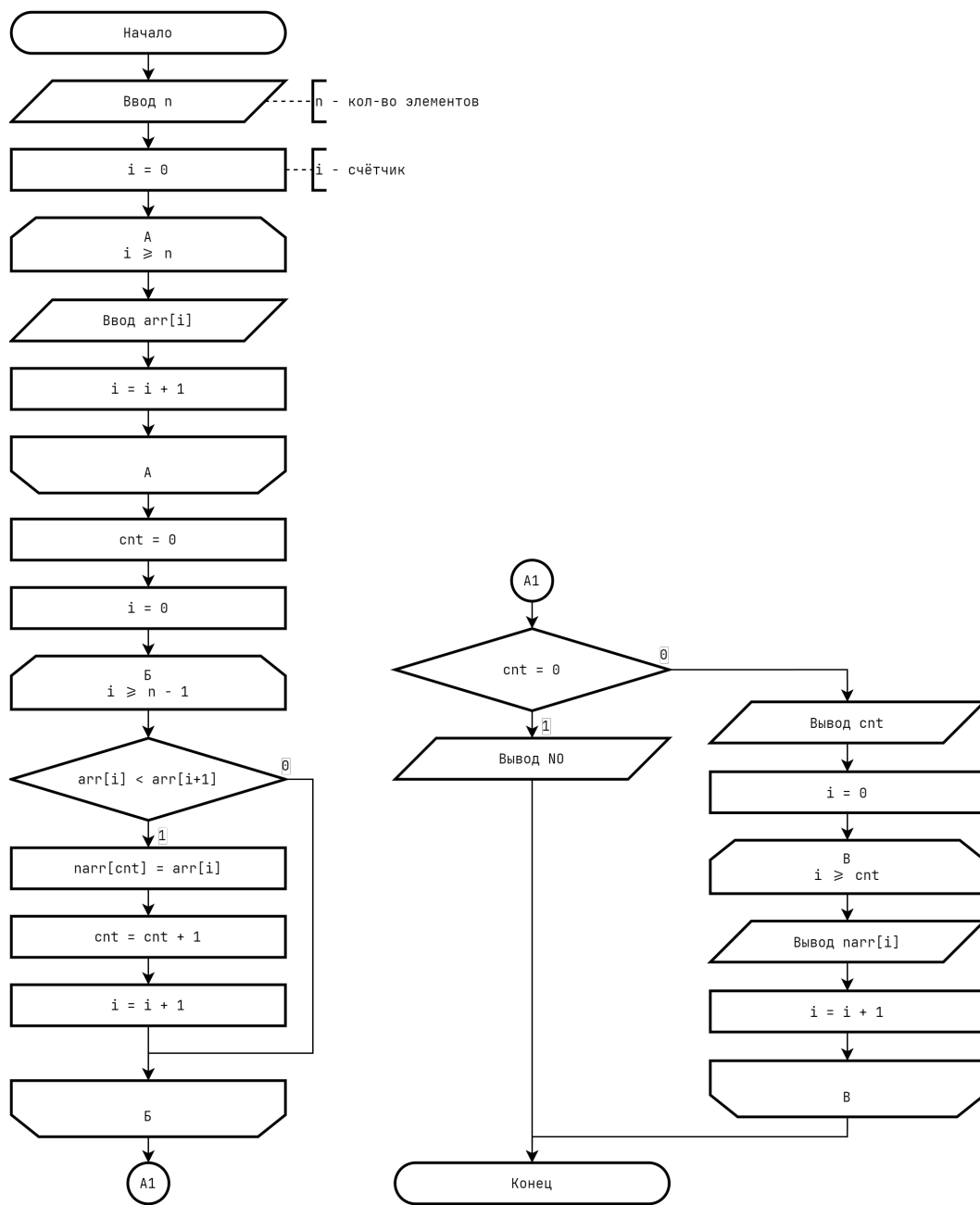


Рисунок 4 - Схема алгоритма Задания 4.

Задание 5.

Схема алгоритма Задания 5 представлена на Рисунке 5.

Исходный код представлен в Приложении А5.

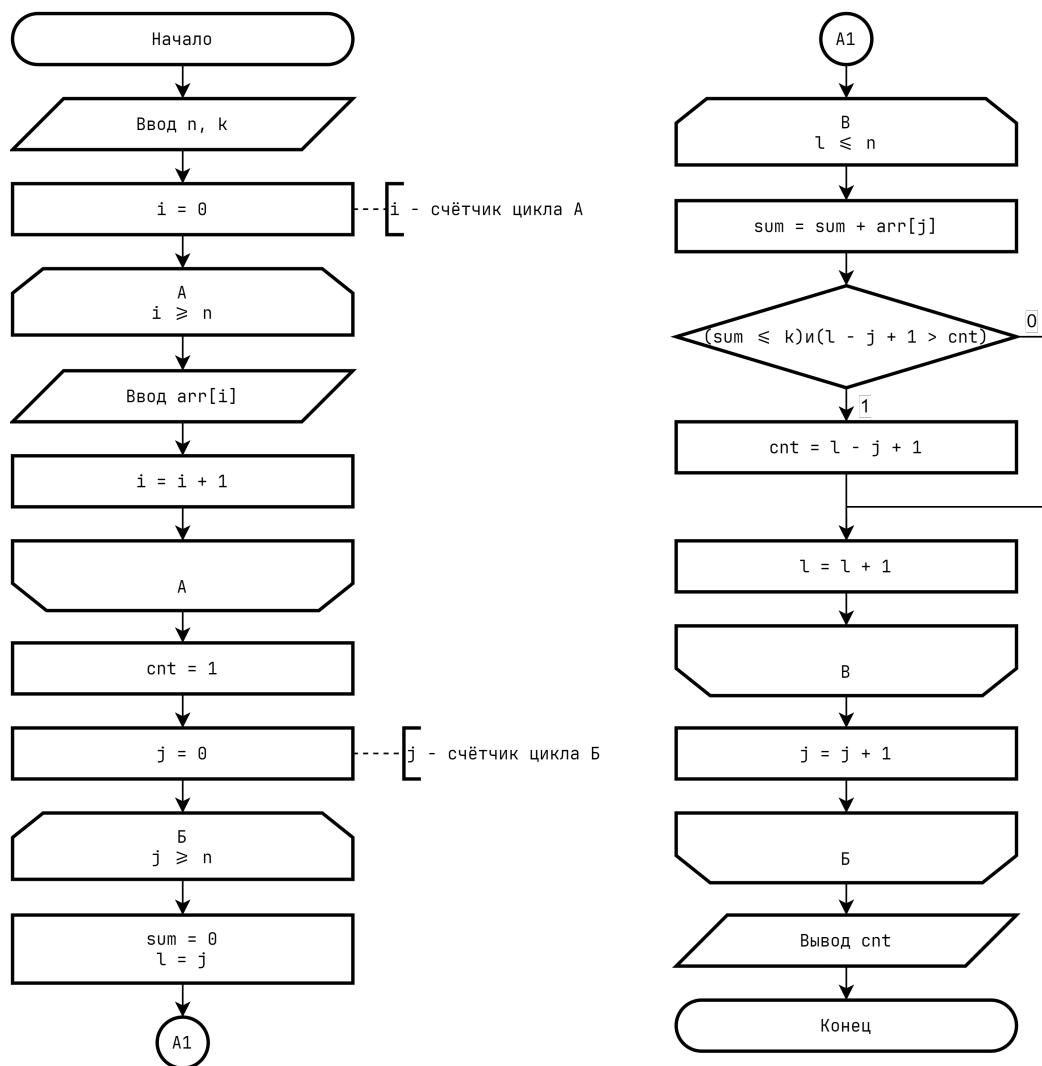


Рисунок 5 - Схема алгоритма Задания 5.

Задание 6.

Схема алгоритма Задания 6 представлена на Рисунке 6.

Исходный код представлен в Приложении А6.

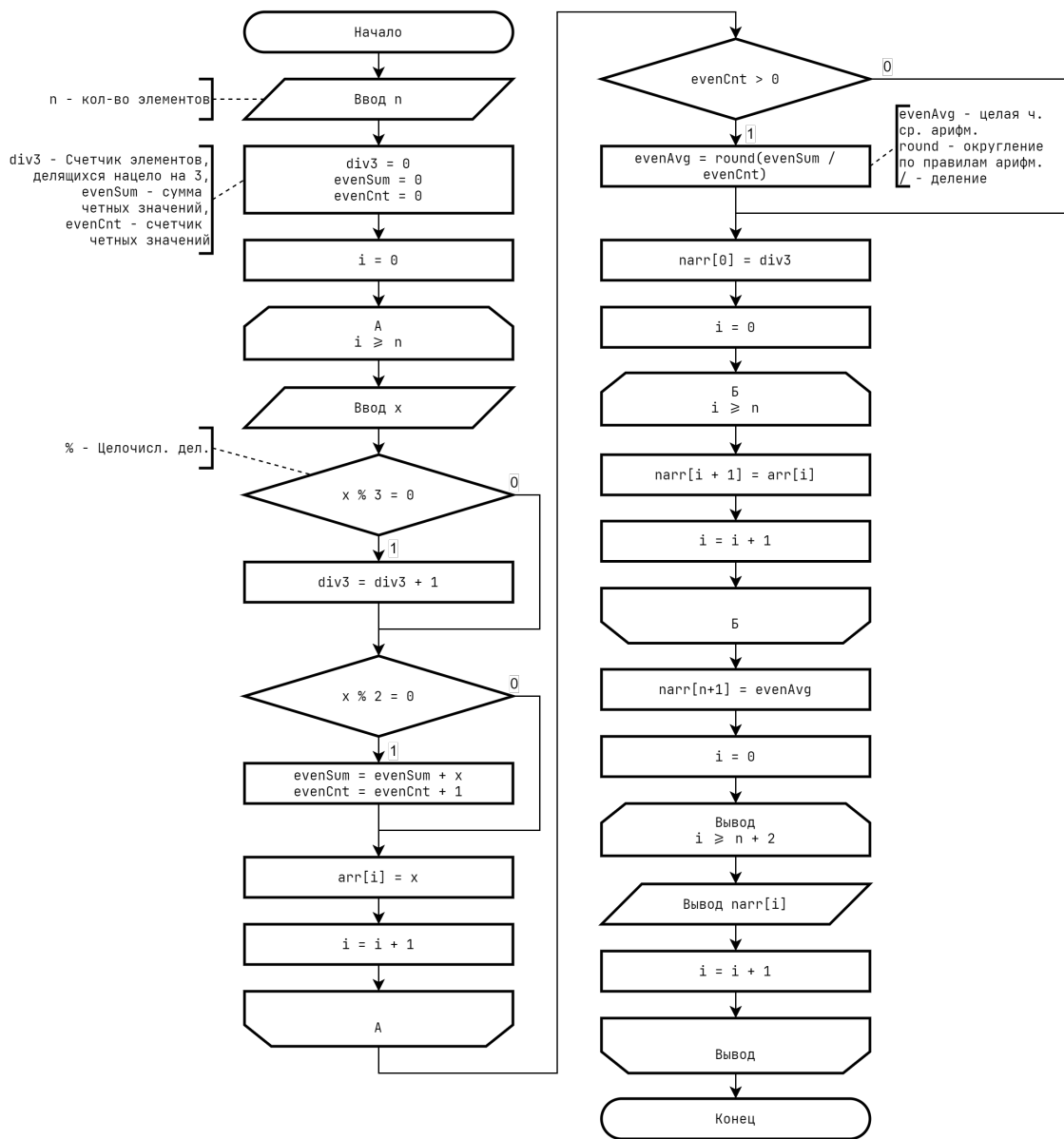


Рисунок 6 - Схема алгоритма Задания 6.

Задание 7.

Схема алгоритма Задания 7 представлена на Рисунке 7.

Исходный код представлен в Приложении А7.

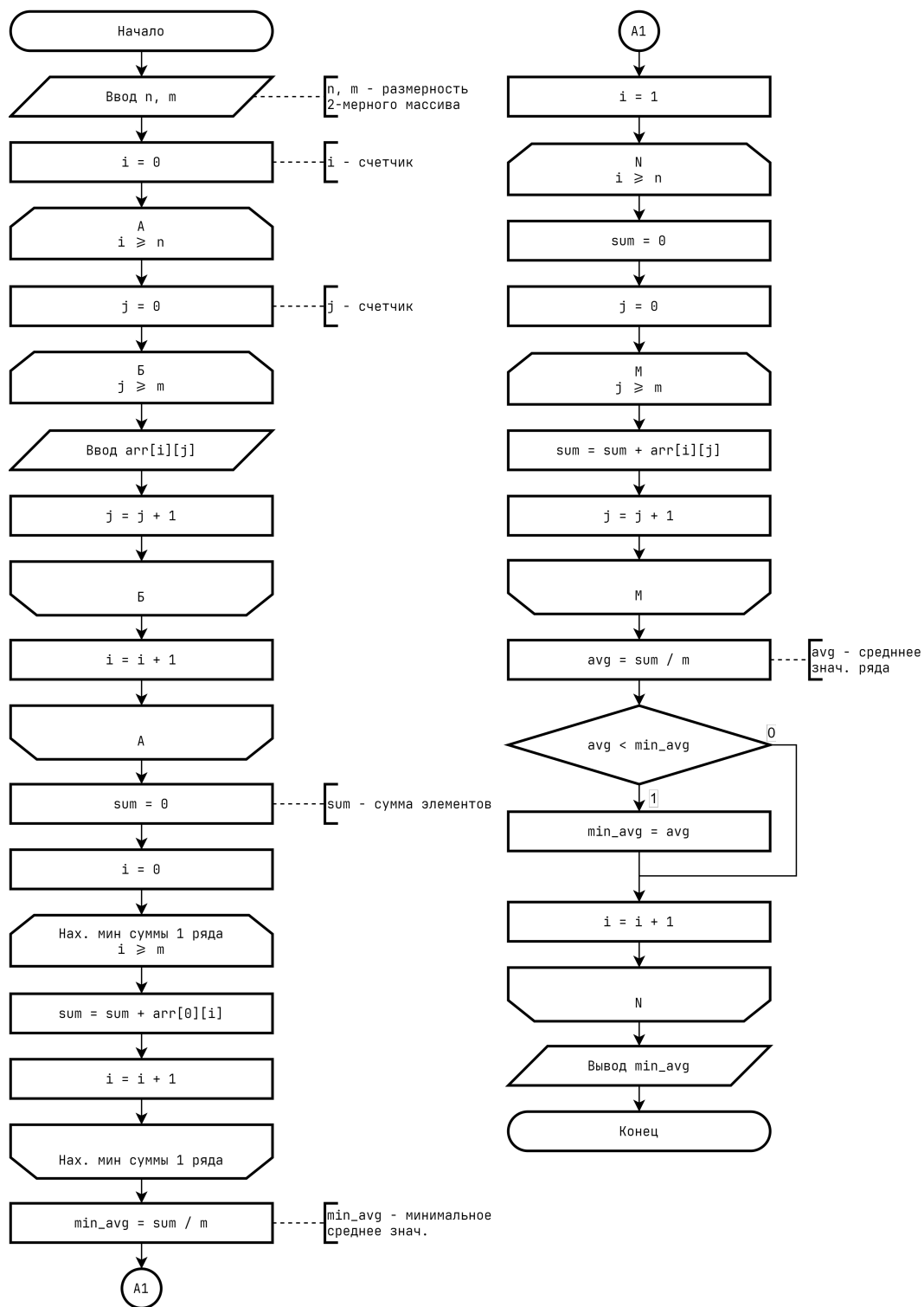


Рисунок 7 - Схема алгоритма Задания 7.

Задание 8.

Схема алгоритма Задания 8 представлена на Рисунке 8.

Исходный код представлен в Приложении А8.

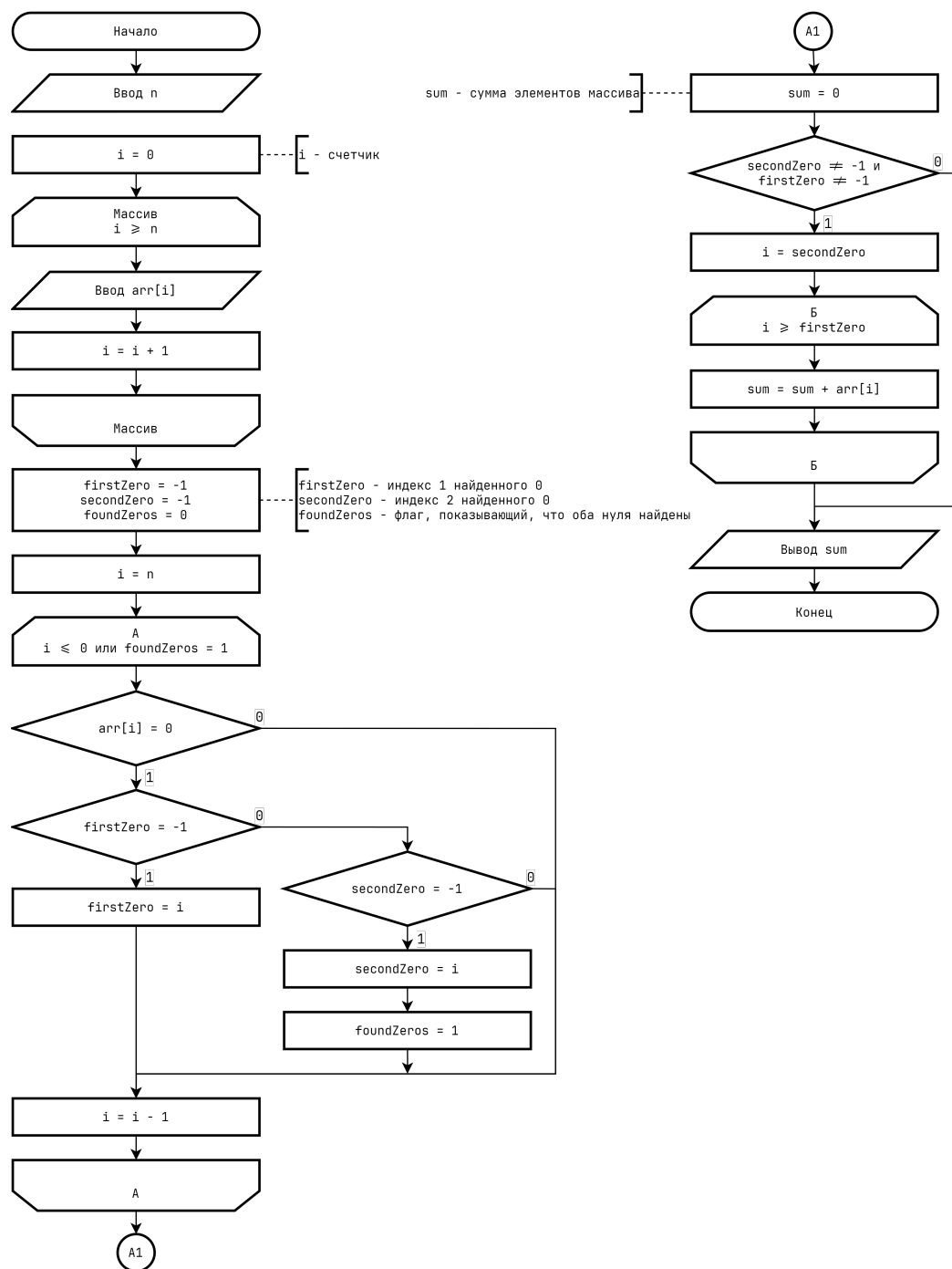


Рисунок 8 - Схема алгоритма Задания 8.

Задание 9.

Схема алгоритма Задания 9 представлена на Рисунке 9.

Исходный код представлен в Приложении А9.

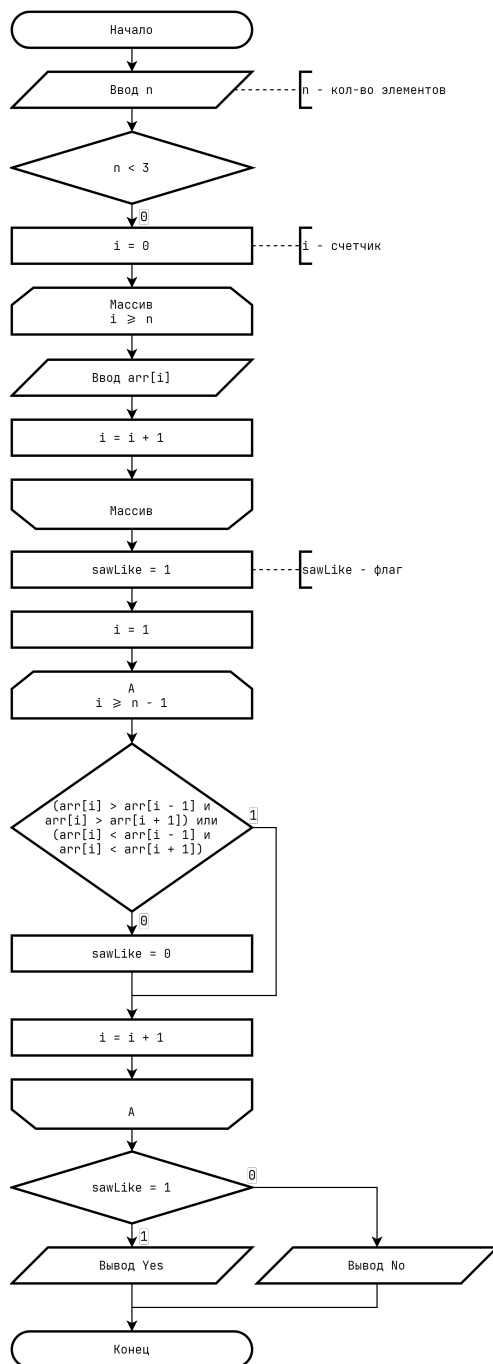


Рисунок 9 - Схема алгоритма Задания 9.

Задание 10.

Схема алгоритма Задания 10 представлена на Рисунке 10.

Исходный код представлен в Приложении А10.

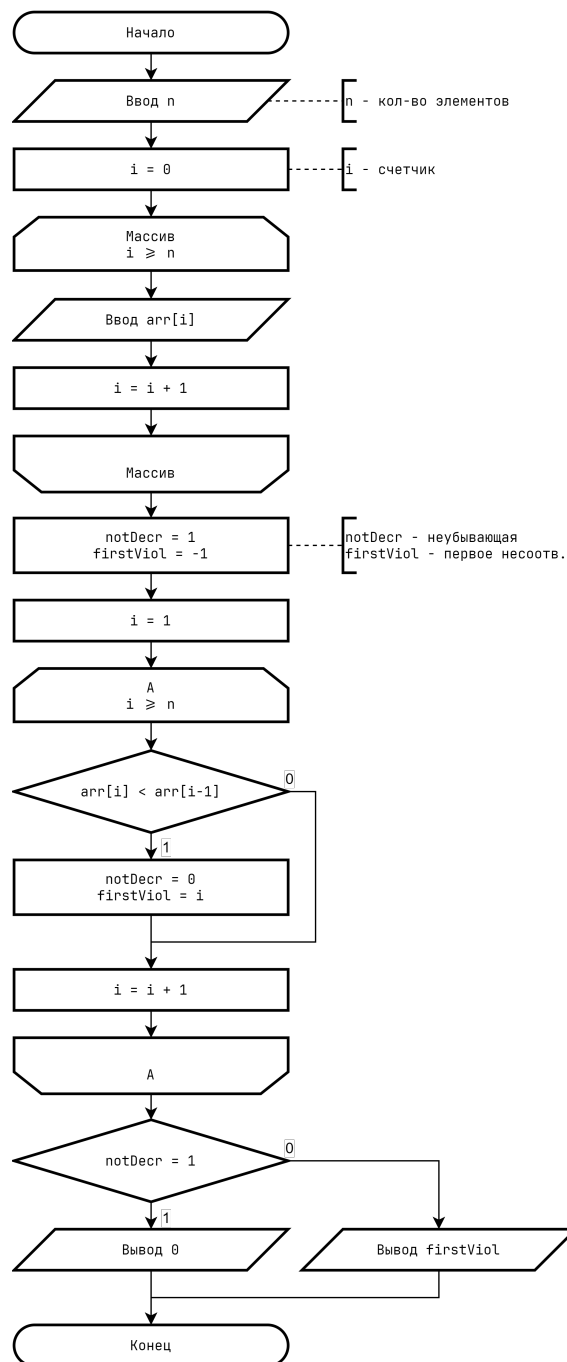


Рисунок 10 - Схема алгоритма Задания 10.

Вывод

В ходе выполнения лабораторной работы были изучены и реализованы алгоритмы обработки одномерных массивов и матриц на языках программирования Pascal и C. Работа включала решения задач различного уровня сложности, направленных на развитие навыков алгоритмического мышления, обработки данных и анализа результатов.

Приложение А1. Исходный код для Задания 1.

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

int main(){
    int n;
    scanf("%d", &n);
    int *arr = (int*)malloc(sizeof(int)*n);
    int *narr = (int*)malloc(sizeof(int)*n);

    for(int i = 0; i < n; i++){
        scanf("%d", &arr[i]);
    }

    for(int i = 0; i < n; i++){
        if(arr[i]==0){
            if(i == 0){
                narr[0] = arr[n-1] + arr[n-2];
            } else if (i == 1){
                narr[1] = arr[0] + arr[n-1];
            } else narr[i] = arr[i-2] + arr[i-1];
        } else narr[i] = arr[i];
    }
```

```

}

for(int i = 0; i < n; i++){
    printf("%d ", narr[i]);
}
free(arr);
free(narr);
return 0;
}

```

Приложение A2. Исходный код для Задания 2.

```

var
    n, i, j, prev: integer;
    arr: array of integer;
    flag, nigg: boolean;
begin
    readln(n);
    SetLength(arr, n);

    nigg := True;

    for i := 0 to n-1 do begin
        read(arr[i]);
    end;

    j := 1;
    flag := True;
    prev := arr[0];
    if arr[0] < 0 then nigg := False;

    while (arr[j] >= 0) do begin
        if (arr[j] >= prev) then flag := False;
    end;
end;

```

```

        prev := arr[j];
        j := j + 1
    end;

    if (flag and nigg) = True then writeln('Yes')
    else writeln('No')
end.

```

Приложение А3. Исходный код для Задания 3.

```

#include <stdio.h>
#include <stdlib.h>

int main(){
    int n;
    scanf("%d", &n);

    int *arr = (int*)malloc(sizeof(int)*n);

    for (int i = 0; i < n; i++){
        scanf("%d", &arr[i]);
    }

    int cnt = 0, res = 0, ind = 0;

    for (int i = 0; i < n; i++){
        if (arr[i] == 1){
            cnt++;
            if (res <= cnt){
                ind = i-cnt+1;
                res = cnt;
            }
        } else cnt = 0;
    }
}

```

```

}
printf("%d %d", ind, res);
free(arr);
return 0;
}

```

Приложение А4. Исходный код для Задания 4.

```

var
    n, i, newSize: Integer; // Кол-во элементов в массиве, счетчик, размер нового массива
    arr: array of Integer; // Исходный массив
    newArr: array of Integer; // Новый массив
begin
    readln(n);
    SetLength(arr, n);
    SetLength(newArr, n);
    for i := 0 to n - 1 do begin
        read(arr[i]);
    end;

    newSize := 0;

    for i := 0 to n - 2 do begin
        if arr[i] < arr[i + 1] then begin
            newArr[newSize] := arr[i];
            newSize := newSize + 1;
        end;
    end;

    if newSize = 0 then begin
        writeln('NO');
    end
    else begin

```

```

        writeln(newSize);
        for i := 0 to newSize - 1 do begin
            write(newArr[i], ' ');
        end;
        writeln;
    end;
end.

```

Приложение А5. Исходный код для Задания 5.

```

#include <stdio.h>
#include <stdlib.h>

int main(){
    int n, k;
    scanf("%d %d\n", &n, &k);

    int *arr = (int*)malloc(sizeof(int)*n);

    for (int i = 0; i < n; i++){
        scanf("%d", &arr[i]);
    }

    int cnt = 1;

    for (int i = 0; i < n; i++){
        int sum = 0;

        for(int j = i; j < n; j++){
            sum += arr[j];

            if ((sum <= k) && (j - i + 1 > cnt)){
                cnt = j - i + 1;
            }
        }
    }
}

```

```

        }
    }
}
printf("%d", cnt);
free(arr);
return 0;
}

```

Приложение А6. Исходный код для Задания 6.

```

uses
    Math;
var
    n, i, div3, evenSum, evenCount, evenAvg: Integer;
    arr, narr: array of Integer;
begin
    ReadLn(n);

    div3 := 0;
    evenSum := 0;
    evenCount := 0;

    SetLength(arr, n);
    SetLength(narr, n + 2);

    for i := 0 to n - 1 do
        begin
            ReadLn(arr[i]);

            if arr[i] mod 3 = 0 then
                Inc(div3);
            if arr[i] mod 2 = 0 then
                begin

```

```

        evenSum := evenSum + arr[i];
        Inc(evenCount);
    end;
end;

if evenCount > 0 then
    evenAvg := Round(evenSum / evenCount)
else
    evenAvg := 0;
narr[0] := div3;

for i := 0 to n - 1 do
    narr[i + 1] := arr[i];
narr[n + 1] := evenAvg;

for i := 0 to n + 1 do
    Write(narr[i], ' ');
WriteLn;
end.

```

Приложение А7. Исходный код для Задания 7.

```

#include <stdio.h>
#include <stdlib.h>

int main(){
    int n, m;
    scanf("%d %d\n", &n, &m);

    int **arr = (int**)malloc(sizeof(int*)*n);
    for (int i = 0; i < n; i++) {
        arr[i] = (int*)malloc(m * sizeof(int));
    }
}

```



```

for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        scanf("%d", &arr[i][j]);
    }
}

double sum = 0;
for (int j = 0; j < m; j++) {
    sum += arr[0][j];
}
double min_avg = sum / m;

for (int i = 1; i < n; i++) {
    sum = 0;
    for (int j = 0; j < m; j++) {
        sum += arr[i][j];
    }
    double avg = sum / m;
    if (avg < min_avg) {
        min_avg = avg;
    }
}

printf("%.2f\n", min_avg);

for (int i = 0; i < n; i++) {
    free(arr[i]);
}
free(arr);
}

```

Приложение A8. Исходный код для Задания 8.

```
var
  n, i, firstZero, secondZero, sum: integer;
  arr: array of Integer;
begin
  readln(n);
  SetLength(arr, n);
  for i := 0 to n - 1 do
    read(arr[i]);

  firstZero := -1;
  secondZero := -1;
  for i := n - 1 downto 0 do
    begin
      if arr[i] = 0 then
        begin
          if firstZero = -1 then
            firstZero := i
          else if secondZero = -1 then
            begin
              secondZero := i;
              Break;
            end;
          end;
        end;
      end;
    end;
  sum := 0;
  if (secondZero <> -1) and (firstZero <> -1) then
    begin
      for i := secondZero + 1 to firstZero - 1 do
        sum := sum + arr[i];
      end;
      writeln(sum);
    end.
end.
```

Приложение А9. Исходный код для Задания 9.

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

int main(){
    int n;
    scanf("%d", &n);

    if (n < 3) {
        printf("No");
        return 0;
    }

    int *arr = (int*)malloc(sizeof(int)*n);

    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int sawLike = 1;

    for (int i = 1; i < n - 1; i++) {
        // не условие
        if (!((arr[i] > arr[i - 1] && arr[i] > arr[i + 1]) || (arr[i] < arr[i - 1] && arr[i] < arr[i + 1]))) {
            sawLike = 0;
            break;
        }
    }

    if (sawLike) {
        printf("Yes");
    } else {
        printf("No");
    }
}
```

```

    }

    free(arr);
    return 0;
}

```

Приложение A10. Исходный код для Задания 10.

```

var
  n, i, firstViol: integer;
  arr: array of real;
  notDecr: boolean;
begin
  readln(n);
  SetLength(arr, n);
  for i := 0 to n - 1 do read(arr[i]);
  notDecr := True;
  firstViol := -1;
  for i := 1 to N - 1 do begin
    if arr[i] < arr[i - 1] then begin
      notDecr := False;
      firstViol := i;
      break;
    end;
  end;
  if notDecr then writeln(0)
  else writeln(firstViol);
end.

```