Министерство транспорта РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Российский университет транспорта» (МИИТ)

Лабораторная работа №5

«Работа с подмассивами»

по дисциплине

«Методы программирования»

Выполнил:

студент группы ТКИ-311,

Гаврилин-Кирьянов И.А.

Проверил:

к.т.н., доц. Логинова Л. Н,

к.т.н., доц. Сафронов А. И.

Москва – 2022 г.

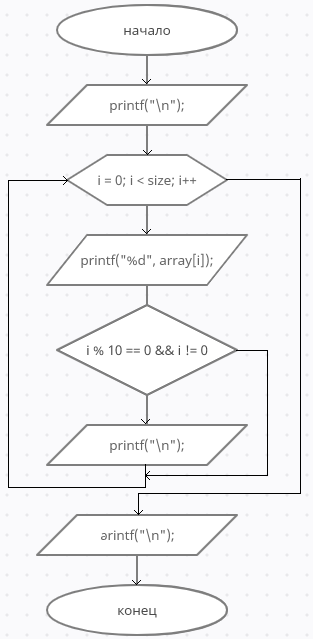
**Цель работы:**

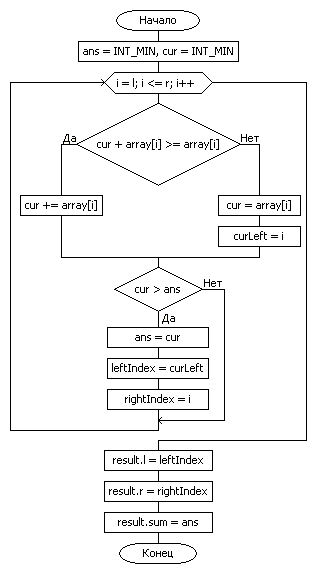
Изучение алгоритма поиска наибольшего подмассива.

**Задача:**

Реализовать программу, выполняющую алгоритм поиска наибольшего подмассива, используя язык программирования С

**Блок-схемы:**





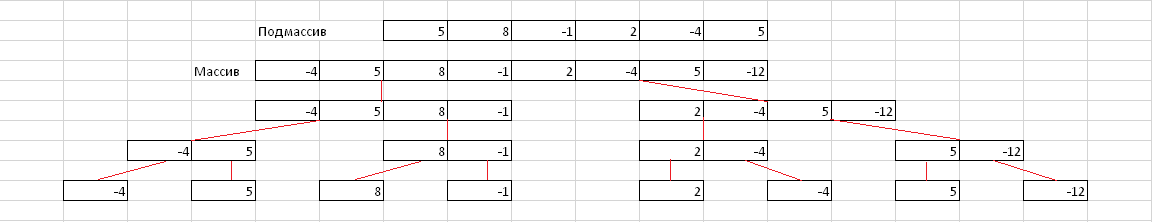
**Тестовые примеры:**

Массив длиной 8, 1000, 10000 и 100000 элементов

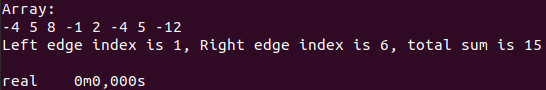
Имеем массив:

-4, 5, 8, -1, 2, -4, 5, -12

Проверим вручную:



Проверим:



**Код программы:**

#include <limits.h>

#include <stdio.h>

int size;

struct subArray

{

int l;

int r;

int sum;

} SUB\_ARRAY, result;

void find\_max(int array[], int l, int r)

{

int ans = INT\_MIN, cur = INT\_MIN;

int curLeft, leftIndex, rightIndex;

for (int i = l; i <= r; i++)

{

if (cur + array[i] >= array[i])

cur += array[i];

else

{

cur = array[i];

curLeft = i;

}

if (cur > ans)

{

ans = cur;

leftIndex = curLeft;

rightIndex = i;

}

}

result.l = leftIndex;

result.r = rightIndex;

result.sum = ans;

return;

}

void print\_array(int array[size], int size)

{

printf("\n");

for (int i = 0; i < size; i++)

{

printf("%d ", array[i]);

if (i % 15 == 14)

{

printf("\n");

}

}

printf("\n");

return;

}

void main()

{

scanf("%d", &size);

int array[size];

for (int i = 0; i < size; i++)

{

scanf("%d", &array[i]);

}

printf("Array:");

print\_array(array, size);

find\_max(array, 0, size - 1);

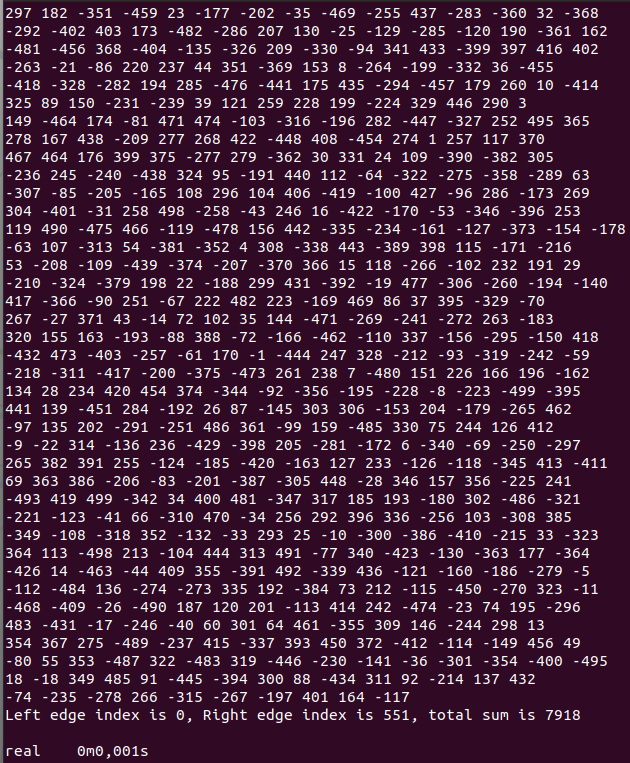
printf("Left edge index is %d, Right edge index is %d, total sum is %d\n", result.l, result.r, result.sum);

return;

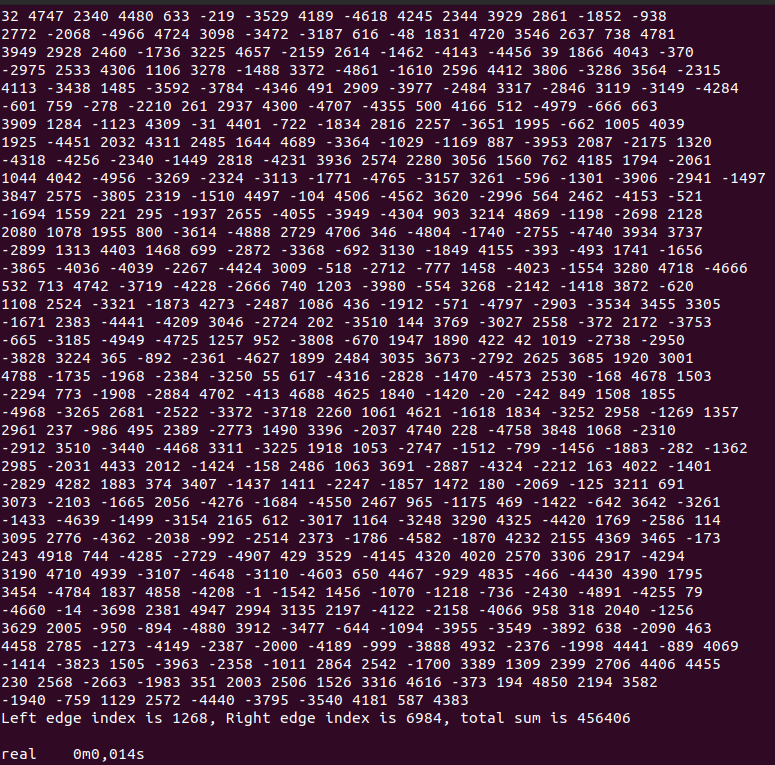
}

**Результаты программы:**

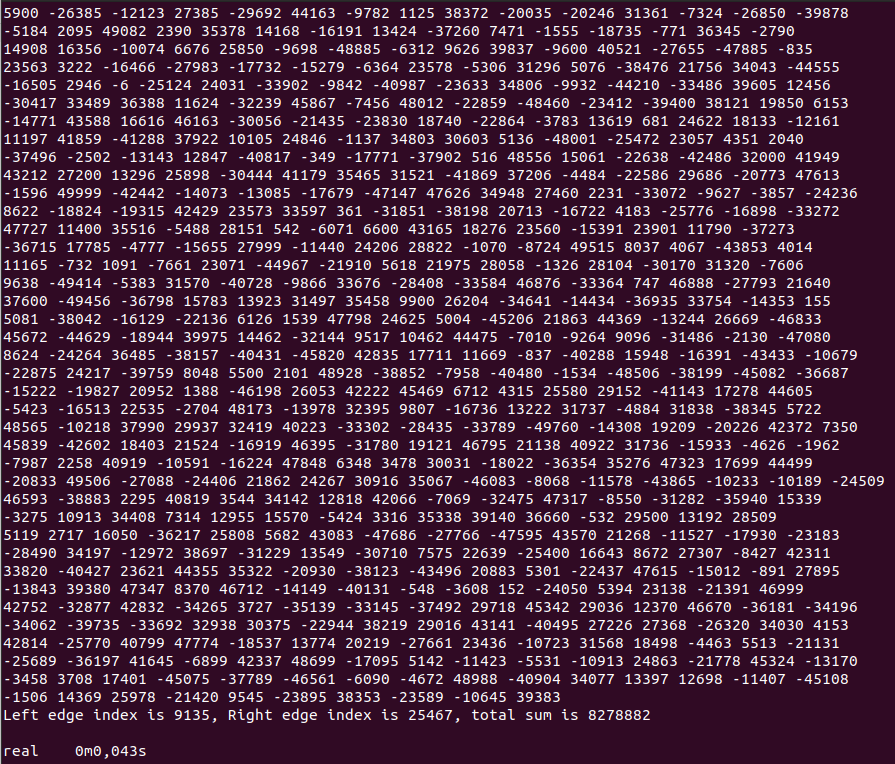
1000 элементов:



10000 элементов:



100000 элементов:



**Вывод:**

В ходе работы был реализован алгоритм поиска наибольшего подмассива. Данный алгоритм имеет сложность O(n), что делает его крайне эффективным при достаточно больших массивах.