МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

Преподаватель

канд. техн. наук, доцент

Л.Н. Бариков

Отчет

по лабораторной работе № 8 по дисциплине Информатика

на тему: «Суммирование рядов»

Работу выполнил

студент гр. 4342

С.М.Иванов

Санкт-Петербург 2024 Дан числовой массив b_0 , b_1 ,..., b_{2n-1} . Определить сумму значений элементов массива с нечётными номерами, лежащих между последним элементом с наибольшим среди неположительных элементов значением и последним элементом с положительным значением, имеющим номер меньше, чем n.

код на гитхабе для более удобного просмотра

https://github.com/crimson-catfish/SUAI labs/blob/main/lab6.cpp

Пользователю предлагается ввести половину желаемой длины массива, которая должна быть больше 0.

Программа создает массив целых чисел, длина которого в два раза превышает длину, предоставленную пользователем, и пользователю предлагается ввести целочисленные значения для каждого элемента массива.

Затем программа ищет целое число с самым отрицательным знаком в массиве и сохраняет его значение и индекс. Если целые числа со знаком не найдены, программа печатает «Числа со знаком не найдены!» и снова начинается с шага 1.

Затем программа ищет последнее целое число без знака (больше 0) в первой половине массива и сохраняет его индекс. Если в первой половине массива не обнаружено целых чисел без знака, программа печатает «В первой половине массива не найдено чисел без знака!» и снова начинается с шага 1.

Программа вычисляет индексы левой и правой границ на основе сохраненных индексов самого отрицательного целого числа со знаком и последнего целого числа без знака в первой половине массива. Левый граничный индекс является минимальным из двух индексов, а правый граничный индекс является максимальным из двух индексов. Если индекс левой границы четный, он увеличивается на 1.

Программа вычисляет сумму элементов массива между левым и правым граничными индексами, увеличивая ее на 2, чтобы пропустить каждый второй элемент.

Сумма печатается, и пользователю предлагается ввести «R» для повторного запуска программы или любую другую букву для выхода.

```
Enter half of array length (greater than 0): 5
-1 -4 -6 89 90 -3 10 11 23 11
Left bound index: 1
Right bound index: 3
Sum: 85
Enter "R" to run again, or enter any other letter to quit programm: r
Enter half of array length (greater than 0): 4
89 12 -4 -9 -1 -2 89 67 23 23
Left bound index: 1
Right bound index: 4
Sum: 3
#include <iostream>
#include <cmath>
using namespace std;
int main()
char quitOrRestart = 'r';
do
int n;
cout << "\nEnter half of array length (greater than 0): ";</pre>
cin >> n;
if (n < 1)
cout << "length should be greater than 0";
int b[n * 2];
```

for (int i = 0; i < 2 * n; i++) { cin >> b[i]; }

int max_signed = -INFINITY;

int max_signed_index;
for (int i = 0; i < n * 2; i++)</pre>

{

```
if (b[i] < 0 \&\& b[i] > max_signed)
max_signed = b[i];
max_signed_index = i;
if (max_signed == -INFINITY)
cout << "No signed numbers found!";
continue;
}
int last_before_n_unsigned_index = -1;
for (int i = n - 1; i \ge 0; i--)
if (b[i] > 0) last_before_n_unsigned_index = i;
if (last_before_n_unsigned_index == -1)
cout << "No unsigned numbers found in first half of array!";
continue;
}
int left_bound = min(max_signed_index, last_before_n_unsigned_index);
int right_bound = max(max_signed_index, last_before_n_unsigned_index);
if (left bound % 2 == 0) left bound++;
cout << "Left bound index: " << left_bound << endl;</pre>
cout << "Right bound index: " << right_bound << endl;</pre>
int sum = 0;
for (int i = left_bound; i <= right_bound; i += 2)</pre>
sum += b[i];
}
cout << "Sum: " << sum;
cout << "\n\nEnter \"R\" to run again, or enter any other letter to quit programm: ";
cin >> quitOrRestart;
} while (quitOrRestart == 'r' | | quitOrRestart == 'R');
```

```
return 0;
}
```