## Інститут спеціального зв'язку та захисту інформації Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського"

Спеціальна кафедра № 5

**ЛАБОРАТОРНА РОБОТА**

з навчальної дисципліни

# “ Алгоритмізація та програмування ”

**Тема**: Обробка масивних даних.

**Варіант № 4**

**Виконав:** студент Жванський Роман Ігорович

### Перевірив: Куліков В.М.

**Київ 2022**

Завдання: В одномірному масиві, що складається з N дійсних чисел, обчислити:

* суму елементів масиву з непарними елементами;
* суму елементів масиву, які розташовані між першим і останнім від’ємними елементами;
* переставити перші M елементів в кінець масиву (M вводиться з клавіатури, M<N).

# Математичне забезпечення

За умовою M(кількількість яку треба перенести в кінець) < n (розмір масиву).

Функція fabs - округлює числа по модулю. Функція fmod використана для ділення дійсних чисел.

Для пошуку чисел які розташовані між першим і останнім від’ємними елементами існує три випадків на які треба зробити перевірку:

1. Від'ємних елементів в масиві немає.
2. Від’ємний елемет 1.
3. На наявність елементів між від'ємними.

Для перестановки елементів також треба виконати перевірку - M < n.

# Алгоритм





# Програма

#include <stdio.h>

#include <math.h> // підключив для функції fabs і fmod

#include <stdlib.h> // підключив для функції calloc

void FillArray(float\* arr, int n); // функція заповнення масиву

float SUMA(float\* arr, int n); // функція суми непарних елементів

float NegativeSum(float\* arr, int n); // суму проміжку від'ємних елементів

void Swap(float\* arr, int n, int M); // функція зміни елементів

void PRINT(float\* arr, int n); // функція для виводу масиву

void DECORATION();// функція для оформлення

int main() {

int n, M;

printf("Input size array (n) ");

scanf("%d", &n); // зчитування розміру масиву

// перевірка на те щоб масив не був меньше 0

if (n <= 0) {

printf("Eror! Not right size!");

return -1;}

float\* arr = calloc(n, sizeof(float)); // віділення пам'яті

DECORATION(); // для оформлення

FillArray(arr, n); // виклив фунції заповнення масиву

DECORATION();

printf("Input M (The number of elements to move to the end) ");

scanf("%d", &M); // зчитування кількості елементів які треба змінити

DECORATION();

// Завдання 1 - вивід непарних елементів

printf("Task 1 (sum unpaired elements ) %.3f\n", SUMA(arr, n));

DECORATION();

// Завдання 2 - вивід суми між непарними елементами

printf("Task 2 (sum elements between negative numbers) ");

printf("%.3f\n", NegativeSum(arr, n));

DECORATION();

// Завдання 3 - змінна елементів місцями

printf("Task 3 (swap elements array) \n");

DECORATION();

printf("Our new array ");

Swap(arr, n, M); // виклив фунції зміни елементів

PRINT(arr, n); // виклик фунції виводу масиву

return 0;

}

void FillArray(float\* arr, int n) {

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

printf("Input elements of array[%d]: ", i);

scanf("%f", &arr[i]);

}

}

float SUMA(float\* arr, int n) {

float suma = 0;

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

// перевірка елементів масиву на парність

// fabs - округлення по модулю fmod - ділення для float

if (!(fabs(fmod(arr[i], 2)) < 1)) {

suma += arr[i];

}

}

return suma;

}

float NegativeSum(float\* arr, int n) {

int begin = -1, end = -1;

float suma = 0;

// пошук першого від'ємного елементу

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

if (arr[i] < 0) {

begin = i + 1;

break;

}

}

// перевірка на наявність від'ємних елементів в масиві

// якщо begin рівне -1 то значить що від'ємних елементів в масиві немає

if (begin == -1) {

printf("- Error! not negative elements ");

return suma;

}

// перевірка на наявність більше 1 від'ємного елемента

// якщо begin рівне n то значить що перший від'ємний елемент є останім в масиві

if (begin == n) {

printf("- Error! not enougt negative elements ");

return suma;

}

// пошук останього від'ємного елементу( з кінця )

for (int i = n - 1; i > begin; i--) {

if (arr[i] < 0) {

end = i - 1;

break;

}

}

// перевірка на наявність елементів між від'ємними

// якщо end рівне -1 то значить що другий від'ємний елемент відсутній, або є сусіднім з першим

if (end == -1) {

printf("- Error! no elements betwen negative elements or no second negative element ");

return suma;

}

// пошук суми між крайніми від'ємними елеменами не включаючи їх

for (int i = begin; i <= end; i++) {

suma += arr[i];

}

return suma;

}

void Swap(float\* arr, int n, int M) {

// тимчасовий масив для заповнення елементами які треба перенести в кінець

float\* temp = calloc(M, sizeof(float));

// перевірка на M < n

if (n < M) {

printf("- Error! M > n\n");

}

else if(M!=n) {

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

// для запису перших M елементів в тимчасовий масив

if (i < M)

temp[i] = arr[i];

// для зміщення інших елементів на початок

else

arr[j++] = arr[i];

}

// запис перших M елементів в кінець масиву

for (int i = n - M, j = 0; j < M; i++, j++) {

arr[i] = temp[j];

}

}

}

void PRINT(float\* arr, int n) {

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

printf("%.3f ", arr[i]);

}

}

void DECORATION() {

printf("---------------------------------------------------\n");// для оформлення

}

# Тест

**Завдання 1**(суму елементів масиву з непарними елементами)

|  |  |  |  |
| --- | --- | --- | --- |
| Данні | Очікуваний результат | Результат програми | Висновок |
| Розмір масиву - 5.Числа в масиві від 1 до 5 | 9 | 9 | Вірно |
| Розмір масиву - 0 | Eror! Not right size! | Eror! Not right size! | Вірно |

**Завдання 2** **(**суму елементів масиву, які розташовані між першим і останнім від’ємними елементами**)**

|  |  |  |  |
| --- | --- | --- | --- |
| Данні | Очікуваний результат | Результат програми | Висновок |
| Розмір масиву - 5.Числа в масиві від 1 до 5 | Error! not negative elements | Error! not negative elements | Вірно |
| Розмір масиву - 5. Числа в масиві: -1 1 1 1 1 | Error! no elements betwen negative elements or no second negative element | Error! no elements betwen negative elements or no second negative element | Вірно |
| Розмір масиву - 5. Числа в масиві: 1 1 1 1 -1 | Error! not enougt negative elements | Error! not enougt negative elements | Вірно |
| Розмір масиву - 5. Числа в масиві: 1 1 -1 -1 1 | Error! no elements betwen negative elements or no second negative element | Error! no elements betwen negative elements or no second negative element | Вірно |

**Завдання 3(**переставити перші M елементів в кінець масиву**)**

|  |  |  |  |
| --- | --- | --- | --- |
| Данні | Очікуваний результат | Результат програми | Висновок |
| Розмір масиву - 5.Числа в масиві від 1 до 5 M = 6 | Error! M > n | Error! M > n | Вірно |
| Розмір масиву - 5.Числа в масиві від 1 до 5 M = 3 | 4 5 1 2 3 | 4 5 1 2 3 | Вірно |
| Розмір масиву - 5.Числа в масиві від 1 до 5 M = 5 | 1 2 3 4 5 | 1 2 3 4 5 | Вірно |

**Висновок**

В процесі лабораторної роботи було сторенно програму, яка має три різні функції обробки одновимірного масиву: SUMA, NegativeSum, SWAP. За допомогою даних функцій реалізовано наступний функціонал:

- знаходження суми непарних елементів одновимірного масиву;

- знаходження суми елементів між першим та останім від’ємними елементами масиву;

- перемещення перших M елементів масиву в кінець даного масиву.

Під час тестування програми було виявлено ряд недоліків. У фінальній версії програми ці недоліки усунуті та додано перевірки на правильність введеня даних.