|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |
| Институт искусственного интеллекта |
| Кафедра Технологий Искусственного Интеллекта |

Практическая работа № 7

по дисциплине

«Процедурное программирование»

Обучающийся: Погосян С. А.

Группа: КВБО-07-23

Руководитель *Яковлев Д. А*

Москва 2023

**Тема: «Массивы как аргументы функций»**

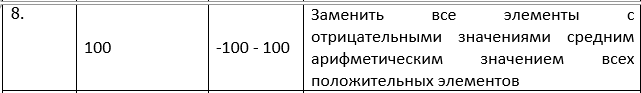
**Цель практической работы:**

Целью данной практической работы освоить на практике работу с функциями, получающими массивы в качестве аргументов.

**Описание работы:**

* Указатели. Типизированные указатели.
* Указатели и массивы.
* Адресная арифметика.
* Динамическое выделение памяти

**Задание:**



**Код программы:**

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#define N 100

double get\_total(double \*ar) {

    double total = 0;

    for (int i =0; i < N; i++) {

        if (ar[i] > 0) {

            total += ar[i];

        }

    }

    return total;

}

int get\_amount(double \*ar) {

    int amount = 0;

    for (int i =0; i < N; i++) {

        if (ar[i] > 0) {

            amount += 1;

        }

    }

    return amount;

}

int main()

{

    srand(time(NULL));

    double \*ar;

    ar = (double \*)malloc(N \* sizeof(double));

    for (int i = 0; i < N; ++i) {

        ar[i] = rand() % (100 + 100 + 1) - 100;

        printf("%3.2lf ", ar[i]);

    }

    printf("\n\n");

    double total = get\_total(ar);

    int amount = get\_amount(ar);

    double average = total / (double)(amount);

    printf("Total = %lf Amount = %d Average = %lf\n\n", total, amount, average);

    for (int i = 0; i <N; ++i) {

        if (ar[i] < 0) {

            ar[i] = average;

        }

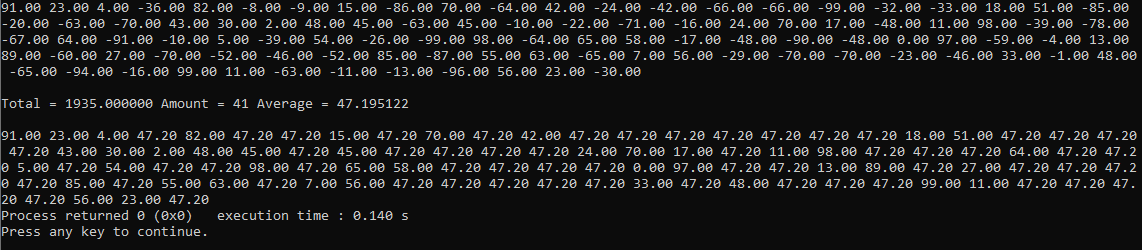
        printf("%3.2lf ", ar[i]);

    }

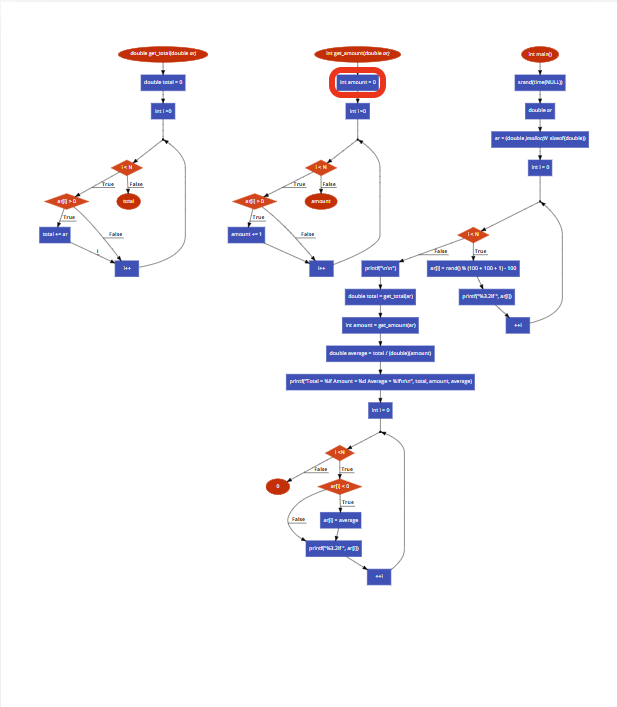
    return 0;

}

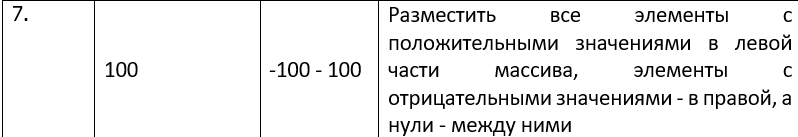
**Результат работы программы:**

****

**Алгоритм в виде блок-схемы:**

****

**Задание:**



**Код программы:**

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#define N 100

int to\_left\_side(int \*ar, int \*left\_ar, int \*count\_left) {

    int index = 0;

    for (int i = 0; i < N; ++i) {

        if (ar[i] > 0) {

            left\_ar[index] = ar[i];

            index += 1;

            \*count\_left = \*count\_left + 1;

        }

    }

    return 0;

}

int to\_right\_side(int \*ar, int \*right\_ar, int \*count\_right) {

    int index = 0;

    for (int i = 0; i < N; ++i) {

        if (ar[i] < 0) {

            right\_ar[index] = ar[i];

            index += 1;

            \*count\_right = \*count\_right + 1;

        }

    }

    return 0;

}

int main() {

    srand(time(NULL));

    int \* ar;

    int \*left\_ar;

    int \*right\_ar;

    int \*middle\_ar;

    int count\_left = 0;

    int count\_right = 0;

    int value = 0;

    ar = (int \*)malloc(N \* sizeof(int));

    left\_ar = (int \*)malloc(N \* sizeof(int));

    right\_ar = (int \*)malloc(N \* sizeof(int));

    printf("Initial Array:\n ");

    for (int i = 0; i < N; ++i) {

        value = rand() % (100 + 100 + 1) - 100;

        if (value == 1) {

            ar[i] = 0;

        } else {

            ar[i] = value;

        }

        printf("%d ", ar[i]);

    }

    to\_left\_side(ar, left\_ar, &count\_left);

    to\_right\_side(ar, right\_ar, &count\_right);

    left\_ar = (int \*)realloc(left\_ar, sizeof(int) \* count\_left);

    right\_ar = (int \*)realloc(right\_ar, sizeof(int) \* count\_right);

    middle\_ar = (int \*)calloc(N - (count\_right + count\_left), sizeof(int));

    putchar('\n');

    printf("Result Array:\n");

    for (int i = 0; i < count\_left; ++i) {

        printf("%3d ", left\_ar[i]);

    }

    for (int i = 0; i < (N - (count\_left + count\_right)); ++i) {

        printf("%3d ", middle\_ar[i]);

    }

    for (int i = 0; i < count\_right; ++i) {

        printf("%3d ", right\_ar[i]);

    }

    free(ar);

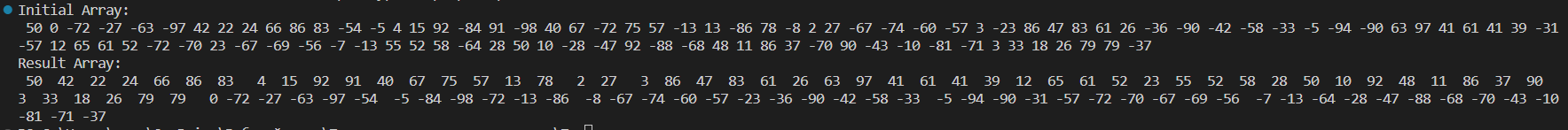
    free(left\_ar);

    free(right\_ar);

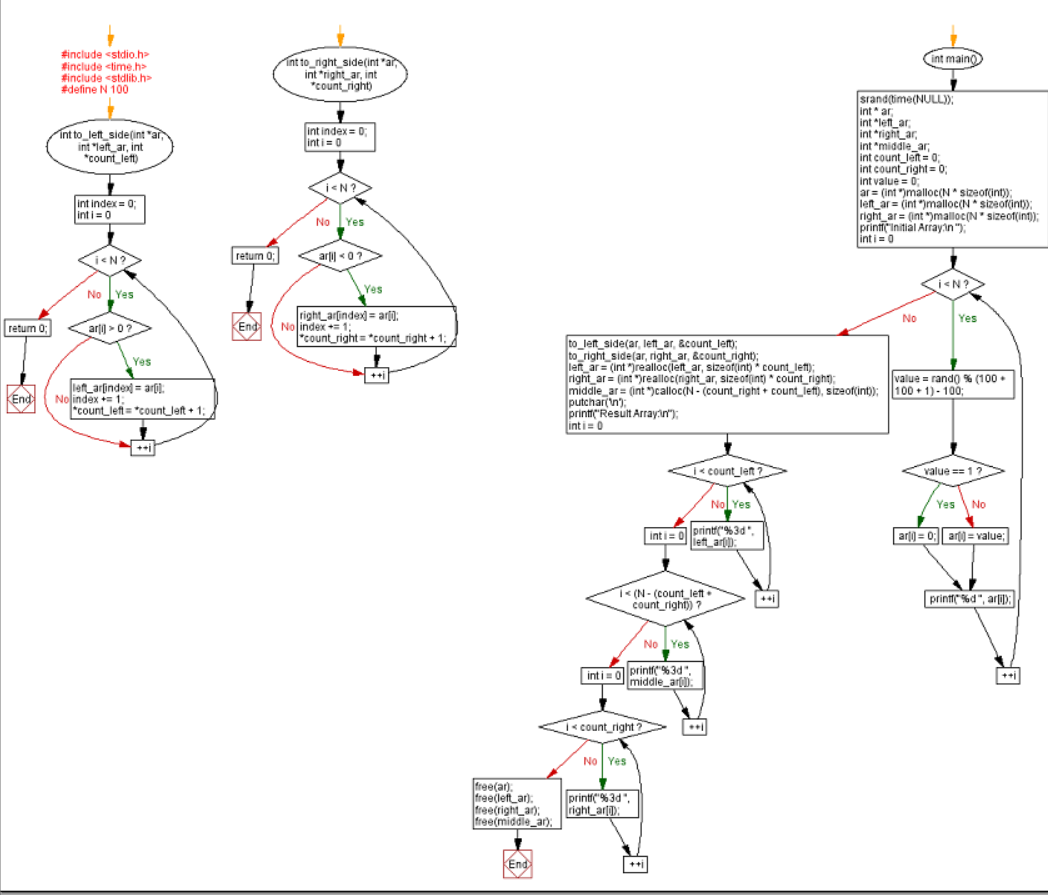
    free(middle\_ar);

}

**Результат работы программы:**



**Алгоритм в виде блок-схемы:**

****