

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Кафедра «Информатика»
кафедра

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ

Алгоритмы сортировки массивов
Тема

Преподаватель

Р. Ю. Царев

подпись, дата

инициалы, фамилия

Студент

КИ19-17/16 031939175

номер группы, зачетной
книжки

подпись, дата

А. Д. Непомнящий

инициалы, фамилия

Красноярск 2020

1 Цель работы

Изучение некоторых прямых алгоритмов сортировки массивов.

2 Задачи

Написать программу, реализующую алгоритмы сортировки массива с помощью включения и с помощью выбора.

Предъявлены следующие требования к выполнению работы.

1. Строгое соответствие программы и результатов ее работы с полученным заданием.
2. Самостоятельное тестирование и отладка программы.
3. Устойчивость работы программы при любых воздействиях, задаваемых пользователем через интерфейс программы.
4. Предоставление демонстрационного примера и исходного текста программы для защиты.
5. Предоставление отчета по практическому заданию, содержащего описание реализованного алгоритма, программы, результатов работы программы (отчет необходимо загрузить на сайт курса).

3 Описание реализованного алгоритма

Реализованы алгоритмы сортировки массивов с помощью включения и с помощью выбора.

4 Описание программы

Для решения задачи была написана программа на языке С. Ниже приведен листинг кода.

Листинг 1 – Код программы

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

enum Constants
{
    INPUT_SIZE = 100
};

void SwapInt(int* a, int* b)
{
    int box;
    box = *b;
    *b = *a;
    *a = box;
}

int CycleInputInt(char* stringToOutput, bool(* pChecker)(int))
{
    int number;
    int position;
    char input[INPUT_SIZE];

    while (true)
    {
        printf("%s\n", stringToOutput);
        fflush(stdout);
        char* fgetsRet = fgets(input, INPUT_SIZE, stdin);
        if (fgetsRet == NULL)
        {
            printf("Wrong format!\n");
            continue;
        }
        int inputLength = strlen(input) - 1;
        input[inputLength] = '\0';
        int sscanfRet = sscanf(input, "%d%n", &number, &position);
        if (position != inputLength)
        {
            printf("Wrong format!\n");
            continue;
        }
    }
}
```

Продолжение листинга 1

```
        }
        if (pChecker && !pChecker(number))
        {
            printf("Wrong format!\n");
            continue;
        }
        if (sscanfRet == 1) break;
        printf("Wrong format!\n");
    }
    return number;
}

bool ArraySizeInputChecker(int arraySize)
{
    return arraySize > 0;
}

bool AnyIntInputChecker(int _)
{
    return true;
}

typedef struct
{
    int* content;
    int size;
} DynArray;

void PrintArray(DynArray* object)
{
    for (int i = 0; i < object->size; i++)
    {
        printf("%d ", object->content[i]);
    }
    printf("\n");
}

void ArrayCopy(DynArray* origin, DynArray* object)
{
    free(object->content);
    object->size = origin->size;
```

Продолжение листинга 1

```
    object->content = (int*) malloc(object->size * sizeof(int));
    for (int i = 0; i < object->size; i++)
    {
        object->content[i] = origin->content[i];
    }
}

void InsertionSort(DynArray* object)
{
    for (int i = 0; i < object->size; i++)
    {
        int box;
        box = object->content[i];
        int j;
        j = 0;
        while (object->content[j] < box)
        {
            j++;
        }
        for (int k = i - 1; k >= j; k--)
        {
            object->content[k + 1] = object->content[k];
        }
        object->content[j] = box;
    }
}

void SelectionSort(DynArray* object)
{
    for (int i = 0; i < object->size; i++)
    {
        int jBox;
        jBox = i;
        int box;
        box = object->content[i];
        for (int j = i; j < object->size; j++)
        {
            if (object->content[j] < box)
            {
                jBox = j;
                box = object->content[j];
            }
        }
        object->content[jBox] = object->content[i];
        object->content[i] = box;
    }
}
```

Окончание листинга 1

```
        }
    }
    SwapInt (&(object->content[i]), &(object->content[jBox]));
}
}

int main()
{
    DynArray object;
    object.size = CycleInputInt("Enter size of array", ArraySizeInputChecker);
    object.content = (int*) malloc(object.size * sizeof(int));
    printf("Enter elements, one by one\n");
    for (int i = 0; i < object.size; i++)
    {
        object.content[i] = CycleInputInt("Enter next element",
                                           AnyIntInputChecker);
    }

    printf("Origin array:\n");
    PrintArray(&object);
    printf("Insertion sorted:\n");
    DynArray coppy;
    ArrayCopy(&object, &coppy);
    InsertionSort(&coppy);
    PrintArray(&coppy);
    printf("Selection sorted:\n");
    ArrayCopy(&object, &coppy);
    SelectionSort(&coppy);
    PrintArray(&coppy);

    free(object.content);
    free(coppy.content);
}
```

5 Результаты работы программы

На следующем рисунке приведен скриншот с результатами работы программы.

```
Enter size of array
6
Enter elements, one by one
Enter next element
19
Enter next element
20
Enter next element
19
Enter next element
0
Enter next element
-2
Enter next element
-1
Origin array:
19 20 19 0 -2 -1
Insertion sorted:
-2 -1 0 19 19 20
Selection sorted:
-2 -1 0 19 19 20

Process finished with exit code 0
|
```

Рисунок 1 – Результаты работы программы