

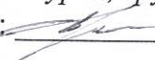
Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего профессионального образования  
«СЕВЕРО-ВОСТОЧНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ ИМЕНИ  
М.К.АММОСОВА»  
ЧУКОТСКИЙ ФИЛИАЛ

## КОНТРОЛЬНАЯ РАБОТА

по дисциплине: Структуры и алгоритмы обработки данных

Выполнил: Гунькин А.А.  
(Фамилия И.О.)

студент 4 курса, группы ИВТ 30-12

Подпись: 

Преподаватель: к.ф.м.н. Звонилов В.И.

Оценка: 3,0 Дата 5.3.16

Подпись: 

Анадырь, 2016

# КОНТРОЛЬНАЯ РАБОТА 1

Гуныкин Артем Андреевич

1.

Используя в качестве массива набор из 8 букв своих фамилии, имени, отчества, определить на каждом шаге в методе прямого выбора номера перемещаемых элементов.

■  
А Р Т Е Г У Н Ъ  
↕

А | Р ■ Т ■ Е ■ Г ■ У Н Ъ  
↔

А Г | Т ■ Е ■ Р У Н Ъ  
↔

А Г Е | Т ■ Р ■ У ■ Н ■ Ъ  
↔

А Г Е Н | Р ■ У Т Ъ  
↕

А Г Е Н Р | У ■ Т ■ Ъ  
↔

А Г Е Н Р Т | У ■ Ъ  
↕

А Г Е Н Р Т У | Ъ

2.

Используя в качестве массива набор из 8 букв своих фамилии, имени, отчества, определить на каждом шаге в методе шейкерной сортировки левую и правую границы сортируемой части массива (L и R).

А Р Т Е Г У Н Ь

Н Ь

Н У

Г Н

Г Е

Г Т

Г Р

А Г

А Г | Р Т Е Н У Ь

Р Т

Е Т

Н Т

Т У

У Ь

А ~~Г~~ | Р Е Н | Т У Ь

Е Н

Е Р

Г Е

А Г Е | Р Н | Т У Ь

Н Р

А Г Е | Н | Р Т У Ь

3.

Используя в качестве массива набор из 8 букв своих фамилии, имени, отчества провести 3-сортировку (в методе Шелла).

### Сортировка 3

А Р Т **Е** Г У Н Ъ  
А Р Т Е **Г** У Н Ъ  
А Г Т Е Р **У** Н Ъ  
А Г Т Е Р У **Н** Ъ  
А Г Т Е Р У Н **Ъ**  
А Г Т Е Р У Н Ъ

### Сортировка 2

А Г **Т** Е Р У Н Ъ  
А Г Т **Е** Р У Н Ъ  
А Г Т Е **Р** У Н Ъ  
А Г Р Е Т **У** Н Ъ  
А Г Р Е Т У **Н** Ъ  
А Г Н Е Р У Т **Ъ**  
А Г Н Е Р У Т Ъ

### Сортировка 1

А **Г** Н Е Р У Т Ъ  
А Г **Н** Е Р У Т Ъ  
А Г Н **Е** Р У Т Ъ  
А Г Е Н **Р** У Т Ъ  
А Г Е Н Р **У** Т Ъ  
А Г Е Н Р У **Т** Ъ  
А Г Е Н Р Т У **Ъ**  
А Г Е Н Р Т У Ъ

4.

Используя в качестве массива набор из 10 букв своих фамилии, имени, отчества, построить пирамиду и с помощью пирамиды отсортировать массив.

1 2 3 4 5 6 7 8 9 10

А Р Т Е М А Г У Н Ъ

А Г У Н Ъ

М А Г У Н Ъ

Е М А Г У Н Ъ

Т Е М А Г У Н Ъ

А Е М Т Г У Н Ъ

Р А Е М Т Г У Н Ъ

Е А Р М Т Г У Н Ъ

Е А Н М Т Г У Р Ъ

А Е А Н М Т Г У Р Ъ



5.

Провести слияние двух упорядоченных списков. В качестве элементов первого списка взять буквы фамилии (полностью), в качестве элементов второго списка взять буквы имени (полностью).

Гунькин Артем

Составим первый список из упорядоченных букв фамилии: Г И К Н Н У Ъ

Составим второй список из упорядоченных букв имени: А Е М Р Т

Г И К Н<sup>1</sup> Н<sup>2</sup> У Ъ



А Г Е И К М Н<sup>1</sup> Н<sup>2</sup> Р Т У Ъ

6.

Провести быстрый поиск (2 версии) буквы "Е" (русс.) в массиве из 15 букв своих фамилии, имени, отчества.

15 букв ФИО – Г У Н Ъ К И Н А Р Т Е М А Е Е

Версия 1

А А Г Е Е Е И К М Н Н Р Т У Ъ

А А Г Е Е Е И

Версия 2

А А Г Е Е Е И К М Н Н Р Т У Ъ

А А Г Е Е Е И К

А А Г Е

Г Е

Е



7.

Построить хэш-таблицу методом квадратичных проб для всех букв своих фамилии, имени, отчества.

Возьмем все буквы фамилии, имени и отчества: Гунькин Артем Андреевич

Различных букв всего 14. Это буквы: Г У Н Ъ К И А Р Т Е М Д В Ч

Поэтому выберем размер хэш-таблицы  $m = 17$  (простое число).

Определим номера символов строки:

|   |    |    |    |    |   |   |    |    |   |    |   |   |    |
|---|----|----|----|----|---|---|----|----|---|----|---|---|----|
| Г | У  | Н  | Ь  | К  | И | А | Р  | Т  | Е | М  | Д | В | Ч  |
| 4 | 20 | 14 | 29 | 11 | 9 | 1 | 17 | 19 | 6 | 13 | 5 | 3 | 24 |

Для каждого символа вычисляем его хэш-номер (или последовательность хэш-номеров, если потребуется) и в соответствии с вычисленным номером заносим символ в хэш-таблицу.

$$г: h_0 = 4 \bmod 17 = 4$$

$$у: h_0 = 20 \bmod 17 = 3$$

$$н: h_0 = 14 \bmod 17 = 14$$

$$ь: h_0 = 29 \bmod 17 = 12$$

$$к: h_0 = 11 \bmod 17 = 11$$

$$и: h_0 = 9 \bmod 17 = 9$$

$$а: h_0 = 1 \bmod 17 = 1$$

$$р: h_0 = 17 \bmod 17 = 0$$

$$т: h_0 = 19 \bmod 17 = 2$$

$$е: h_0 = 6 \bmod 17 = 6$$

$$м: h_0 = 13 \bmod 17 = 13$$

$$д: h_0 = 5 \bmod 17 = 5$$

$$в: h_0 = 3 \bmod 17 = 3$$

$$h_1 = 3 + 1 = 4$$

$$h_2 = 4 + 3 = 7$$

$$ч: h_0 = 24 \bmod 17 = 7$$

$$h_1 = 7 + 1 = 8$$

Хэш-таблица выглядит следующим образом:

|   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| р | а | т | у | г | д | е | в | ч | и |    | к  | ь  | м  | н  |    |    |

## Контрольная работа 2

14. Написать процедуру бинарного поиска заданного целого значения в упорядоченном (по возрастанию или убыванию) массиве (x) целых чисел из n элементов. Подсчитать число сравнений.

### Описание решения

- Создаем массив целых чисел из восьми элементов.
- Выводим массив на экран.
- Вводим с клавиатуры элемент для поиска.
- Вызываем функцию для поиска введенного элемента.
- Функция принимает массив, количество его элементов, элемент для поиска и переменную для сохранения в ней количество сравнений (по ссылке). Функция реализует бинарный поиск элемента. Если элемент найден, то возвращает номер элемента, иначе – число -1.
- Выводим номер элемента и число сравнений на экран.

### Текст программы

```
#include <iostream.h>
#include <conio.h>

// Функция принимает упорядоченный массив x размером n
// и возвращает номер элемента num (бинарный поиск)
// В переменной cmp возвращает число сравнений
int Find(int *x, int n, int num, int& cmp)
{
    int L = 0; // Левая граница поиска
    int R = n - 1; // Правая граница поиска
    int m; // Середина
    while (L <= R) {
        m = (L + R) / 2;
        cmp++;
        if (x[m] == num)
            return m;
        cmp++;
        if (x[m] < num)
            L = m + 1;
    }
}
```

```

        else
            R = m - 1;
    }
    return -1; // Возвращаем -1, если элемент не найден
}

void main()
{
    int n = 8; // Размер массива
    int x[] = {3, 6, 8, 10, 31, 55, 67, 323}; // Массив
    cout << "Массив чисел:" << endl;
    for (int i = 0; i < n; i++) // Выводим массив
        cout << x[i] << " ";
    cout << endl;
    int num; // Элемент для поиска
    cout << "Введите элемент для поиска: ";
    cin >> num;
    int cmp = 0; // Число сравнений
    int k = Find(x, n, num, cmp);
    if (k != -1)
        cout << "Номер элемента в массиве: " << k + 1 << endl;
    else
        cout << "Элемент в массиве не найден" << endl;
    cout << "Число сравнений: " << cmp << endl;

    getch();
}

```

**Результаты работы программы**



DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TASK

D:\>task

Массив чисел:

3 6 8 10 31 55 67 323

Введите элемент для поиска: 67

Номер элемента в массиве: 7

Число сравнений: 5

D:\>task

Массив чисел:

3 6 8 10 31 55 67 323

Введите элемент для поиска: 36

Элемент в массиве не найден

Число сравнений: 6