Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и Информационных технологий

Кафедра Информатики

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6**

Преподаватель \_\_\_\_\_\_\_\_ Пересунько П. В.

подпись, дата инициалы, фамилия

Студент КИ15–17Б, 031510065 \_\_\_\_\_\_\_\_\_\_\_ Радионов Т.В.

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2016

**1 Цель работы**

Реализовать скрипт, выполняющий следующие задачи: программа принимает от пользователя элементы целочисленного одномерного массива, а также целочисленное значение для поиска, затем осуществляет поиск элемента двумя любыми известными методами, кроме прямого (линейного) поиска, и выводит результат на экран. Также необходимо реализовать защиту ввода от некорректных данных пользователя.

**2 Исходные тексты программ**

#!/bin/bash

declare -a array

### Считывание массива

function FillingArray()

{

local check=0

while [ "$check" = "0" ]

do

check=1

# Ввод массива с клавиатуры

echo -n "Напишите массив чисел через пробел: "

read -a array

# Проверка элементов на целочисленность

for (( i = 0 ; i < ${#array[@]} ; i++ ))

do

[ ${array[i]} -ge 0 -o ${array[i]} -lt 0 ] || check=0

done

done

}

### Поиск элемента по ключу

function FindElement()

{

# Ввод ключа с клавиатуры

declare -i key

echo -n "Введите элемент, который хотите найти: "

read key

# Бинарный поиск элемента

for (( i = 0 ; i < ${#array[@]} ; i++ ))

do

local l=0

local u=${#array[@]}

local m

local flag=0

while [ "$flag" = "0" ]

do

flag=1

let "m = (l + u) / 2"

if [[ "${array[m]}" < "$key" ]]; then

let "l = m + 1"

elif [[ "${array[m]}" > "$key" ]]; then

let "u = m - 1"

elif [[ "${array[m]}" = "$key" ]]; then

break

fi

if [[ "$l"<="$u" ]]; then

flag=0

fi

if [[ "$u"<="0" ]]; then

m=-1

break

fi

m=-1

done

done

echo "Позиция искомого элемента (бинарный поиск): $m"

# Интерполяционный поиск элемента

for (( i = 0 ; i < ${#array[@]} ; i++ ))

do

local l=0

local u=${#array[@]}

let "u = u - 1"

local m

local flag=0

while [ "$flag" = "0" ]

do

flag=1

let "m = l + ( ( u - l) \* ( $key - ${array[l]} ) ) / ( ${array[u]} - ${array[l]} )"

if [[ "${array[m]}" < "$key" ]]; then

let "l = m + 1"

elif [[ "${array[m]}" > "$key" ]]; then

let "u = m - 1"

elif [[ "${array[m]}" = "$key" ]]; then

break

fi

if [[ "${array[l]}"<"$key" && "${array[u]}">="$key" ]]; then

flag=0

fi

m=-1

done

done

echo "Позиция искомого элемента (интерполяционный поиск): $m"

}

### Тело скрипта

again="yes"

while [ "$again" = "yes" ]

do

FillingArray

FindElement

echo -n "Вы желаете продолжить? (yes/no): "

read again

done

# Освобождение памяти

unset array

**3 Примеры работы программ**

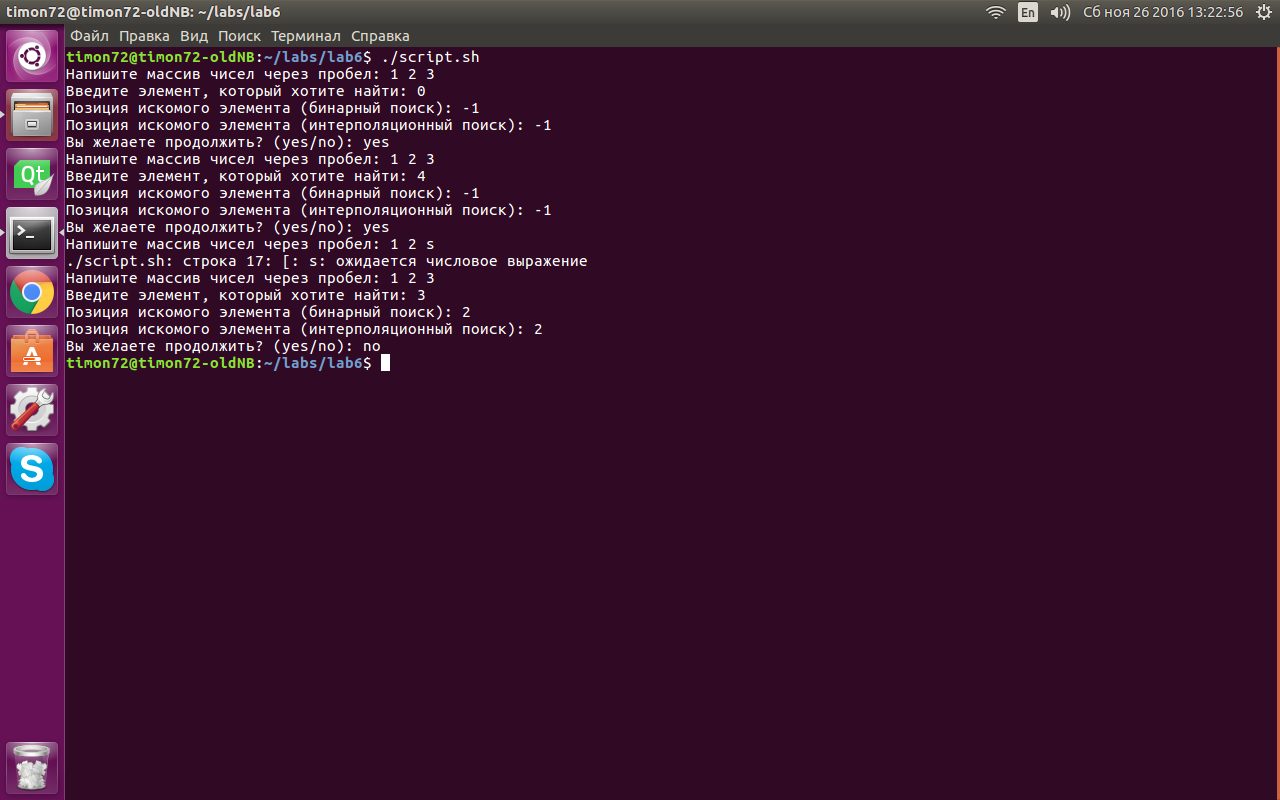


Рисунок 1