**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

**Кафедра информационных систем**

**ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1  
по курсу: Теория информационных процессов и систем**

**Парадигмы и языки программирования**

**Работу выполнили студенты:**

**Трофимов Влад**

**группа 3511**

**Виноградов Павел**

**группа 3511**

**САНКТ-ПЕТЕРБУРГ**

**2015**

В ходе лабораторной работы был реализован алгоритм бинарного поиска на 4 языках программирования и нарисованы графы управления.

**Исходный код:**

**Язык С:**

#include <stdio.h>

#include <stdlib.h>

int cmpint(int \*pa, int pb)

{

int a = \*pa;

int b = pb;

if (a < b) return -1;

if (a == b) return 0;

if (a > b) return 1;

}

int bsearch(int Array[], int n, int key,int(\*cmp)(int \*, int pb))

{

unsigned left = 1, right = n;

int NotFound = -1;

if (!(Array && n > 0 && key && cmp))

return NotFound;

while (left < right)

{

unsigned m = (left + right) / 2;

if (cmp(Array + m, key) < 0)

left = m + 1;

else

right = m;

}

return (cmp(Array + right, key) == 0) ? right : NotFound;

}

int main(void)

{

int a[] = {1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024};

int n = sizeof(a) / sizeof(\*a) - 1;

int key = 4;

int b = bsearch(a, n, key,cmpint);

printf("Position of %d in [ ", key);

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

{

printf("%d ", a[i]);

}

printf("] is %d\n", b);

key = 0;

b = bsearch(a, n, key, cmpint);

printf("Position of %d in [ ", key);

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

{

printf("%d ", a[i]);

}

printf("] is %d\n", b);

return 0;

}

**Язык Java:**

import java.util.Arrays;

import java.util.List;

public class Bsearch {

public static void main(String[] args) {

List<Integer> list = Arrays.asList(1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024);

Integer[] array = (Integer[])list.toArray();

int value = 4;

int result = new ExtendedArray(array).binarySearch(value);

System.out.println("Position of " + value + " in " + list + " is " + result);

value = 0;

result = new ExtendedArray(array).binarySearch(value);

System.out.println("Position of " + value + " in " + list + " is " + result);

}

}

class ExtendedArray {

private Integer[] data;

public ExtendedArray(Integer[] source) {

this.data = source;

}

public int binarySearch(int value){

int h = data.length - 1;

int l = 0;

while (h >= l) {

int m = l + ((h - l) / 2);

if (data[m] > value) {

h = m - 1;

} else if (data[m] < value) {

l = m + 1;

} else {

return m;

}

}

return -1;

}

}

**Язык Common Lisp:**

(defun binary-search (value array &optional (low 0) (high (1- (length array))))

(if (< high low)

nil

(let ((middle (floor (/ (+ low high) 2))))

(cond ((> (aref array middle) value)

(binary-search value array low (1- middle)))

((< (aref array middle) value)

(binary-search value array (1+ middle) high))

(t middle)))))

(setq array (make-array '(11)

:initial-contents

'(1 2 4 8 16 32 64 128 256 512 1024)))

(setq value 4)

(setq result (binary-search value array))

(format t "Position of ~S~%in ~S~%is ~S~%" value array result)

(setq value 0)

(setq result (binary-search value array))

(format t "Position of ~S~%in ~S~%is ~S~%" value array result)

|  |
| --- |
|  |

**Язык Prolog:**

bin\_search(Elt,List,Result):-

length(List,N), bin\_search\_inner(Elt,List,1,N,Result).

bin\_search\_inner(Elt,List,J,J,J):-

nth(J,List,Elt).

bin\_search\_inner(Elt,List,Begin,End,Mid):-

Begin < End,

Mid is (Begin+End) div 2,

nth(Mid,List,Elt).

bin\_search\_inner(Elt,List,Begin,End,Result):-

Begin < End,

Mid is (Begin+End) div 2,

nth(Mid,List,MidElt),

MidElt < Elt,

NewBegin is Mid+1,

bin\_search\_inner(Elt,List,NewBegin,End,Result).

bin\_search\_inner(Elt,List,Begin,End,Result):-

Begin < End,

Mid is (Begin+End) div 2,

nth(Mid,List,MidElt),

MidElt > Elt,

NewEnd is Mid-1,

bin\_search\_inner(Elt,List,Begin,NewEnd,Result).

**Графы:**

**C**



**Java**



**Common Lisp**



**Prolog**

