Experiment 12: Binary search

Code:

#include<stdio.h>

struct Array

{

int A[10];

int size;

int length;

};

void Display(struct Array arr)

{

int i;

printf("\nElements are\n");

for(i=0;i<arr.length;i++)

printf("%d ",arr.A[i]);

}

void swap(int \*x,int \*y)

{

int temp=\*x;

\*x=\*y;

\*y=temp;

}

int BinarySearch(struct Array arr,int key)

{

int l,mid,h;

l=0;

h=arr.length-1;

while(l<=h)

{

mid=(l+h)/2;

if(key==arr.A[mid])

return mid;

else if(key<arr.A[mid])

h=mid-1;

else

l=mid+1;

}

return -1;

}

int RBinSearch(int a[],int l,int h,int key)

{

int mid=0;

if(l<=h)

{mid=(l+h)/2;

if(key==a[mid])

return mid;

else if(key<a[mid])

return RBinSearch(a,l,mid-1,key);

}

else

return RBinSearch(a,mid+1,h,key);

return -1;

}

int main()

{

struct Array arr1={{2,3,9,16,18,21,28,32,35},10,9};

printf("%d",BinarySearch(arr1,16));

Display(arr1);

return 0;

}

Output:

