program -2

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

void linear\_search(int key,int array[100],int n)

{

int i,loc,flag=1;

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

{

if(key == array[i])

{

loc = i;

printf("The location of Search Key = %d is %d\n",key,loc);

flag=0;

}

}

if(flag==1)

{

printf("The Search Key NOT FOUND in an array\n");

}

}

int binarySearch(int arr[], int l, int r, int x)

{

if (r >= l)

{

int mid = l + (r - l) / 2;

if (arr[mid] == x)

{

return mid;

}

if (arr[mid] > x)

return binarySearch(arr, l, mid - 1, x);

return binarySearch(arr, mid + 1, r, x);

}

return -1;

}

int main()

{

int x, j = 0,flag, n , sort = 0, i, ch;

clock\_t start, end;

int arr[1000];

printf("Enter the number of elements\n");

scanf("%d",&n);

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

{

int no = rand() % n + 1;

arr[i] = no;

}

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

{

printf(" %d ",arr[m]);

}

printf("\n\n");

for(;;)

{

printf("1.Linear Search \n2.Binary Search \n");

scanf("%d",&ch);

switch(ch)

{

case 1:

printf("Enter the element to be Searched : ");

scanf("%d", &x);

start = clock();

linear\_search(x,arr,n);

end = clock();

float ti = ((double)(end - start)/CLOCKS\_PER\_SEC);

printf("Time taken: %f\n\n", ti);

break;

case 2:

j = 0;

flag = 0;

int temp;

while(j < n-1 && sort == 0 )

{

if(arr[j] > arr[j+1])

{

flag=1;

}

j++;

}

if(flag == 1)

{

for (int i = 0; i < n-1; i++)

{

for (j = 0; j < n-i-1; j++)

{

if (arr[j] > arr[j+1])

{

temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

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

{

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

}

sort = 1;

printf("\n\n");

}

printf("Enter the element to be Searched : ");

scanf("%d",&x);

start = clock();

int result = binarySearch(arr, 0, n - 1, x);

end = clock();

if(result == -1)

{

printf("Element Not Found.\n");

}

else

{

printf("Element found at position %d.\n", result + 1);

}

float tm = ((double)(end - start)/CLOCKS\_PER\_SEC);

printf("Time taken: %f\n\n",tm);

break;

default:

exit(0);

}

}

return 0;

}

output:

[shilpa@shilpa-linux ADA\_LAB-PROG-main]$ gcc -o linear\_binary linear\_binary.c

[shilpa@shilpa-linux ADA\_LAB-PROG-main]$ ./linear\_binary

Enter the number of elements

100

84 87 78 16 94 36 87 93 50 22 63 28 91 60 64 27 41 27 73 37 12 69 68 30 83 31 63 24 68 36 30 3 23 59 70 68 94 57 12 43 30 74 22 20 85 38 99 25 16 71 14 27 92 81 57 74 63 71 97 82 6 26 85 28 37 6 47 30 14 58 25 96 83 46 15 68 35 65 44 51 88 9 77 79 89 85 4 52 55 100 33 61 77 69 40 13 27 87 95 40

1.Linear Search

2.Binary Search

1

Enter the element to be Searched : 23

The location of Search Key = 23 is 32

Time taken: 0.000099

1.Linear Search

2.Binary Search

1

Enter the element to be Searched : 14

The location of Search Key = 14 is 50

The location of Search Key = 14 is 68

Time taken: 0.000126

1.Linear Search

2.Binary Search

2

3 4 6 6 9 12 12 13 14 14 15 16 16 20 22 22 23 24 25 25 26 27 27 27 27 28 28 30 30 30 30 31 33 35 36 36 37 37 38 40 40 41 43 44 46 47 50 51 52 55 57 57 58 59 60 61 63 63 63 64 65 68 68 68 68 69 69 70 71 71 73 74 74 77 77 78 79 81 82 83 83 84 85 85 85 87 87 87 88 89 91 92 93 94 94 95 96 97 99 100

Enter the element to be Searched : 1

Element Not Found.

Time taken: 0.000021

1.Linear Search

2.Binary Search

2

Enter the element to be Searched : 20

Element found at position 14.

Time taken: 0.000016

1.Linear Search

2.Binary Search