DS LAB

Week 1: Array Manipulation and Searching Techniques

a. Arrays – DS
Objective: Reverse array elements
https://www.hackerrank.com/challenges/arrays-ds/problem?isFullScreen=true

Without Functions	With functions

#include <stdio.h>
int main() {
int n;
int i,temp=0;
scanf("%d",&n);
int a[n];
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
for(i=0;i<n/2;i++)
{
temp=a[i];
a[i]=a[n-1-i];
a[n-1-i]=temp;
}
for(i=0;i<n;i++)
{
printf("%d\t",a[i]);
}
return 0;
}
	#include <stdio.h>
	
	void inputArray(int n, int a[]) {
	printf("Enter the elements: ");
	for (inti = 0; i< n; i++) {
	scanf("%d", &a[i]);
	}
	}
	
	void reverseArray(int n, int a[]) {
	int temp;
	for (inti = 0; i< n / 2; i++) {
	temp = a[i];
	a[i] = a[n - 1 - i];
	a[n - 1 - i] = temp;
	}
	}
	
	void printArray(int n, int a[]) {
	printf("Reversed array: ");
	for (inti = 0; i< n; i++) {
	printf("%d\t", a[i]);
	}
	printf("\n");
	}
	
	intmain() {
	int n;
	printf("Enter the n value: ");
	scanf("%d", &n);
	
	int a[n];

	inputArray(n, a);
	reverseArray(n, a);
	printArray(n, a);
	
	return 0;
	}

b. Linear Search

Objective: Find the position of number K in the given list

https://www.hackerrank.com/contests/17cs1102/challenges/1-a-linear-search

Without functions	With functions

#include <stdio.h>
 	
intmain() {
int n;
inti,temp=0,key,pos=-1;
scanf("%d",&n);
int a[n];
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
scanf("%d",&key);
for(i=0;i<n;i++)
{
if(a[i]==key)
pos=i;
 	
}
printf("%d",pos);
 	
return 0;
	#include <stdio.h>
	
	intfindKeyPosition(int n, int a[], int key) {
	for (inti = 0; i< n; i++) {
	if (a[i] == key) {
	return i; // Return the position if key is found
	}
	}
	return -1; // Return -1 if key is not found
	}
	
	void inputArray(int n, int a[]) {
	printf("Enter the elements: ");
	for (inti = 0; i< n; i++) {
	scanf("%d", &a[i]);
	}
	}
	
	intmain() {
	int n, key;
	printf("Enter the n value: ");
	scanf("%d", &n);

}	int a[n];
	inputArray(n, a);
	
	printf("Enter the key to find: ");
	scanf("%d", &key);
	
	intpos = findKeyPosition(n, a, key);
	
	printf("Position of key: %d\n", pos);
	
	    return 0;
	}

Write a simple program to read int, float, char and string using scanf() and display using printf() in all the above given platforms.

```
#include <stdio.h>

intmain() {
int integer;
    float floatingPoint;
    char character;
    char string[100];

    // Read values
printf("Enter an integer: ");
scanf("%d", &integer);

printf("Enter a float: ");
scanf("%f", &floatingPoint);

printf("Enter a character: ");
scanf(" %c", &character);

printf("Enter a string: ");
scanf("%s", string);

    // Display values
printf("\nYou entered:\n");
printf("Integer: %d\n", integer);
printf("Float: %f\n", floatingPoint);
printf("Character: %c\n", character);
printf("String: %s\n", string);

    return 0;
}
```

c. Binary Search – Basic
Objective: find index (0-based) of a given key in a sorted array
https://www.hackerrank.com/contests/launchpad-1-winter-challenge/challenges/binary-search-basic

```
#include <stdio.h>

int main() {
    int n;
    int i, key, pos = -1, mid = 0, low = 0;

    scanf("%d", &n);
    int high = n - 1;
    int a[n];

    for (i = 0; i < n; i++) {
        scanf("%d", &a[i]);
    }

    scanf("%d", &key);

    while (low <= high) {
        mid = (low + high) / 2;

        if (a[mid] == key) {
            pos = mid;
            break;
        } else if (key > a[mid]) {
            low = mid + 1;
        } else {
            high = mid - 1;
        }
    }

    printf("%d", pos);

    return 0;
}
```

d. Binary Search – Iterative
Objective: Given queries found in array elements or not.
https://www.hackerrank.com/contests/17cs1102/challenges/1-b-binary-searchiterative

```
#include <stdio.h>

int main()
{
    int n, m;
    scanf("%d%d", &n, &m);
    int a[n];
   for (inti = 0; i< n; i++)
     {
        scanf("%d", &a[i]);
     }
  int  key[m];
   for (inti = 0; i< m; i++)
     {
        scanf("%d", &key[i]);
      }
   for (inti = 0; i< m; i++) {
    int left = 0, right = n - 1, target = key[i];
    int found = 0;

    while (left <= right)
      {
        int mid = (left +right ) / 2;

         if (a[mid] == target)
          {
            found = 1;
            break;
           }
        else if (a[mid] < target)
         {
            left = mid + 1;
         }
       else
        {
            right = mid - 1;
         }
      }

       if (found)
        {
           printf("YES\n");
         }
      else
```

```
        {
         printf("NO\n");
        }
    }

    return 0;
}
```

e.Binary Search – Recursion

Objective: Given queries found in array elements or not.

https://www.hackerrank.com/contests/17cs1102/challenges/1-c-binary-search-recursion

```
#include <stdio.h>

int binarySearch(int arr[], int left, int right, int target)   // Recursive binary search function

{
    if (left <= right)
      {
          int mid = (left +right) / 2;

              if (arr[mid] == target)
                  {
                       return 1;  // Element found
                  }
              else if (arr[mid] < target)
                  {
                      return binarySearch(arr, mid + 1, right, target);
                  }
             else
                 {
                     return binarySearch(arr, left, mid - 1, target);
                  }
         }

    return 0; // Element not found
}

int main()
{
     int n, m;
     scanf("%d%d", &n, &m);
     int a[n];
    for (inti = 0; i< n; i++)
        {
```

```
            scanf("%d", &a[i]);
        }

    for (inti = 0; i< m; i++)
      {
          int key;
         scanf("%d", &key);

         int found = binarySearch(a, 0, n - 1, key);    // Call the recursive binary search function

     if (found)
        {
            printf("YES\n");
        }
      else
        {
            printf("NO\n");
        }
    }

    return 0;
}
```