Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering &Technology

Name :Dumpeta Varshitha . Email :varshithadumpeta@gmail.com
Roll no:23071A1023 Phone :8919552327
Branch :VNRVJIET Department :EIE
Batch :2027 Degree :B.Tech EIE

2023_27_II_EIE 1_Data Structures_lab

DATA STRUCTURES_CODING_WEEK 1

Attempt : 1
Total Mark : 50
Marks Obtained : 40

Section 1 : CODING

1. Problem Statement

Rohit is given the role of designing a programming challenge for an educational platform. The challenge involves creating a program that performs the functionality of a stack data structure, implemented using an array.

The main objective of this challenge is to develop a menu-driven program that empowers users to execute a range of stack operations, which include adding elements to the stack (pushing), removing elements from the stack (popping), and visualizing the current contents of the stack (display).

Write a program to help Rohit design the programming challenge.

Answer

```
//to impliment stack
#include<stdio.h>
#define MAX_SIZE 100
int stack[MAX_SIZE],element,i;
```

```
int top=-1,value;
void push(int value){
   if(top==MAX_SIZE-1){
      return;
   }
   stack[++top]=value;
   printf("Element %d is pushed onto the stack\n",value);
}
   void pop()
   {
      if(top==-1)
      {
         printf("Stack Underflow\n");
         return;
      }
      element=stack[top--];
      printf("Element %d is popped from the stack\n",element);
   }
   void displaystack(){
      if(top==-1){
         printf("Stack is empty\n");
         return;
      }
      printf("Elements in the stack: ");
      for(int i=top;i>=0;i--)
      {
         printf("%d",stack[i]);
      }
      printf("\n");
   }
   int main(){
      int choice,value;
      do
      {
         scanf("%d",&choice);
         switch(choice)
         {
            case 1:scanf("%d",&value);
            push(value);
            break;
            case 2:pop();
            break;
```

```
            case 3:displaystack();
            break;
            case 4:printf("Exiting the program\n");
            break;
            default:printf("Invalid choice\n");
        }
    }while(choice!=4);
    return 0;
}
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Implement a recursive linear search algorithm to find the index of a given element in an array. The program should use recursion to search through the array elements one by one in a linear fashion. The goal is to determine whether the target element is present in the array and, if so, provide its index.

Function specification: recSearch(int arr[], int l, int r, int x).

Answer

```
//you are using GCC
#include<stdio.h>
int recSearch(int arr[],int l,int r,int x)
{
    if(r<1)
    return -1;
    if(arr[l]==x)
    return l;
    if(arr[r]==x)
    return r;
    return recSearch(arr,l+1,r-1,x);
}
int main()
{
    int i,n,x;
    scanf("%d",&n);
    int arr[n];
```

```
    for(i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    scanf("%d",&x);
    int index=recSearch(arr,0,n-1,x);
    if(index!=-1)
    printf("Element %d is present at index%d",x,index);
    else
    printf("Element %d is not present",x);
    return 0;
}
```

Status : Correct **Marks : 10/10**

3. Problem Statement

Implement a program to perform a linear search on an array using non-recursion. The program should take as input the size of the array, the array elements, and the target element to be searched. It will then output whether the target element is present in the array and, if so, provide its index.

Function Specification: linearSearch(int arr[], int n, int x)

Answer

```
// You are using GCC
#include<stdio.h>
int linearSearch(int arr[],int n,int x)
{
    for(int i=0;i<n;i++)
    {
        if(arr[i]==x)
        return i;
    }
    return -1;
    }
    int main()
    {
        int n,x;
```

```
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    scanf("%d",&x);
    int result=linearSearch(arr,n,x);
    if(result!=-1)
    printf("Element %d is present at index %d\n",x,result);
    else
    {
        printf("Element %d is not present in the array\n",x);
    }
    return 0;
}
```

Status : Correct **Marks : 10/10**

4. Problem Statement

You are given a sorted integer array of numbers of length n. Your task is to implement a binary search algorithm using recursion to find a target You are provided with a sorted array of integers with a length of n. Your objective is to implement a recursive binary search algorithm to determine whether a target integer is present in the array.

If the target is found, the algorithm should return the index of the target element; otherwise, it should print a message indicating that the element was not found in the array.

Function Specification: binarySearch(int nums[], int left, int right, int target)

Answer

```
// You are using GCC
#include<stdio.h>
int binarySearch(int nums[],int left,int right,int target){
    if(left>right){
        return -1;
```

```
    }
        int mid=(left+right)/2;

       if(nums[mid]==target){
          return mid;
       }else if(nums[mid]<target){
          return binarySearch(nums,mid+1,right,target);
       }else{
          return binarySearch(nums,left,mid-1,target);
       }
}
    int main(){
       int n;
       scanf("%d",&n);

       int nums[n];
       for(int i=0;i<n;i++){
          scanf("%d",&nums[i]);
       }
       int target;
       scanf("%d",&target);
       int result=binarySearch(nums,0,n-1,target);
       if(result!=-1){
       printf("Element %d is found at index %d\n",target,result);
       }else{
       printf("Element %d is not found in the array\n",target);
}
return 0;
}
```

Status : Wrong ***Marks : 0/10***

5. Problem Statement

You are tasked with implementing a non-recursive binary search algorithm. The program takes as input a sorted array of integers and a target value. The objective is to determine whether the target value is present in the array. If found, the program should provide the index at which the target element is located; otherwise, it should indicate that the element is not present.

Function Specification: binarySearch(int arr[], int l, int r, int x)

Answer

```
// You are using GCC
#include<stdio.h>
void binarySearch(int arr[],int l,int r,int x)
{
    int mid;
    while(l<=r)
    {
        mid=(l+r)/2;
        if(arr[mid]==x)
        {
            printf("Element %d is present at index %d",x,mid);
            break;
        }
        else if(arr[mid]<x)
        l=mid+1;
        else
        r=mid-1;
    }
    if(l>r)
    printf("Element %d is not present in array",x);
}
int main()
{
    int i,arr[15],n,x,f,h;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    scanf("%d",&x);
    f=0;
    h=n-1;
    binarySearch(arr,f,h,x);
}
```

Status : Correct ***Marks : 10/10***