Dr. D.Y.Patil Arts, Commerce & Science College,
Pimpri, Pune-18
S.Y.B.Sc(Comp. Sci) 2024-25
Data Structures and Algorithms – I
Solution of Practical Assignment 1: Searching Algorithms
Set A

a) **Create a random array of n integers. Accept a value x from user and use linear search algorithm to check whether the number is present in the array or not and output the position if the number is present.**

```
// LINEAR SEARCH for integers
#include<stdio.h>
void linearsearch(int a[],int n,int key);
main ()
{
        int a[10], n, i,key ;
        printf ("How many element do you want to enter in array : ");
        scanf ("%d", &n);
        for (i = 0; i < n; i++)
        {
                printf ("\n Enter element %d : ", i + 1);
                scanf ("%d",&a[i]);
        }
        printf ("\nEnter the element to be search :");
        scanf ("%d", &key);
        linearsearch(a,n,key);
}
void linearsearch(int a[10],int n,int key)
{
        int i;
        for (i = 0; i < n; i++)
        {
                if (a[i]==key)
                {
                        printf ("\n%d element found at %d position.", key, i + 1);
                        break;
                }
        }

        if (i == n)
                printf ("\nItem %d not in the list \n", key);
}
/*
[root@localhost ds]# cc linear.c
[root@localhost ds]# ./a.out
How many element do you want to enter in array : 5
 Enter element 1 : 3
 Enter element 2 : 6
 Enter element 3 : 1
```

```
 Enter element 4 : 4
 Enter element 5 : 9
Enter the element to be search :4
4 element found at 4 position.
[root@localhost ds]# ./a.out
How many element do you want to enter in array : 5
 Enter element 1 : 7
 Enter element 2 : 9
 Enter element 3 : 2
 Enter element 4 : 3
 Enter element 5 : 1
Enter the element to be search :4
Item 4 not in the list
*/
```

b) **Accept n values in array from user. Accept a value x from user and use sentinel linear search algorithm to check whether the number is present in the array or not and output the position if the number is present.**

```
//Sentinel  LINEAR SEARCH for integers
#include<stdio.h>
int sentinelsearch(int a[],int n,int key);
main ()
{
        int a[10], n, i,key,result ;
        printf ("How many element do you want to enter in array : ");
        scanf ("%d", &n);
        for (i = 0; i < n; i++)
        {
                printf ("\n Enter element %d : ", i + 1);
                scanf ("%d",&a[i]);
        }
        printf ("\nEnter the element to be search :");
        scanf ("%d", &key);
        result=sentinelsearch(a,n,key);
      if(result==-1)
        printf("\nelement  not found");
      else
        printf("\n%d element found at %d position.",key,result+1);
}
int sentinelsearch(int a[10],int n,int key)
{
        int i=0;
      a[n]=key;
        while(a[i]!=key)
          i++;
        if (i == n)
                return -1;
      else
                return i;
}
```

```
/*
[root@localhost ds]# cc ass1setA2.c
[root@localhost ds]# ./a.out
How many element do you want to enter in array : 5

 Enter element 1 : 2
 Enter element 2 : 4
 Enter element 3 : 8
 Enter element 4 : 1
 Enter element 5 : 3
Enter the element to be search :8
8 element found at 3 position.

[root@localhost ds]# ./a.out
How many element do you want to enter in array : 5
 Enter element 1 : 1
 Enter element 2 : 2
 Enter element 3 : 3
 Enter element 4 : 4
 Enter element 5 : 5
Enter the element to be search :7
element  not found
*/
```

c) **Accept n sorted values in array from user. Accept a value x from user and use binary search algorithm to check whether the number is present in sorted array or not and output the position if the number is present.**

```
// BINARY SEARCH for integers

#include <stdio.h>
void binsearch(int a[],int start, int end,int key);
void display(int [],int);

main ()
{
    int a[20],n, i,key;
    printf ("How many element you want to enter in the array : ");
    scanf ("%d", &n);
       printf("\nEnter the Data in sorted format:");
    for (i = 0; i < n; i++)
    {
        printf ("\nEnter the  element %d :", i + 1);
        scanf ("%d", &a[i]);
    }
    printf ("\nEnter the element to be searched  :");
    scanf ("%d", &key);
    binsearch(a,0,n-1,key);
}
void display(int a[20],int n)
{
```

```
    int i;
    printf("\n");
    for(i=0;i<n;i++)
        printf("\t%d",a[i]);
}

void binsearch(int a[],int start, int end,int key)
{
    int mid;
    mid = (start + end) / 2;
    while (key != a[mid] && start <= end)
    {
        if (key > a[mid])
                start = mid + 1;
        else
                end=mid - 1;
        mid = (start + end) / 2;
    }
    if (key == a[mid])
        printf ("\n%d element found at %d posotion.\n ", key, mid + 1);
    if (start > end)
        printf ("\n%d not found in the list\n ", key);
}
/*
[root@localhost ds]# cc ass1setA3.c
[root@localhost ds]# ./a.out
How many element you want to enter in the array : 5

Enter the Data in sorted format:
Enter the  element 1 :1
Enter the  element 2 :2
Enter the  element 3 :3
Enter the  element 4 :4
Enter the  element 5 :5
Enter the element to be searched  :4
4 element found at 4 posotion.

 [root@localhost ds]# ./a.out
How many element you want to enter in the array : 5

Enter the Data in sorted format:
Enter the  element 1 :1
Enter the  element 2 :2
Enter the  element 3 :3
Enter the  element 4 :4
Enter the  element 5 :5
Enter the element to be searched  :8
8 not found in the list

*/
```

// BINARY SEARCH for integers with sorting function

```
#include <stdio.h>
void binsearch(int a[],int start, int end,int key);
void bubblesort(int [],int);
void display(int [],int);

main ()
{
    int a[20],n, i,key;
    printf ("How many element you want to enter in the array : ");
    scanf ("%d", &n);
    for (i = 0; i < n; i++)
    {
        printf ("\nEnter the  element %d :", i + 1);
        scanf ("%d", &a[i]);
    }
    bubblesort(a,n);
    printf("\nData after Sorting: ");
    display(a,n);
    printf ("\nEnter the element to be searched  :");
    scanf ("%d", &key);
    binsearch(a,0,n-1,key);
}
void bubblesort(int a[20],int n)
{
    int i,j,temp,pass;
    for(pass=1;pass<n;pass++)
    {
        for(i=0;i<n-pass;i++)
        {
                if(a[i]>a[i+1])
                {
                        temp=a[i];
                        a[i]=a[i+1];
                        a[i+1]=temp;

                }
        }
    }
}

void display(int a[20],int n)
{
    int i;
    printf("\n");
    for(i=0;i<n;i++)
        printf("\t%d",a[i]);
}

void binsearch(int a[],int start, int end,int key)
{
```

```
    int mid;
    mid = (start + end) / 2;
    while (key != a[mid] && start <= end)
    {
        if (key > a[mid])
                start = mid + 1;
        else
                end=mid - 1;
        mid = (start + end) / 2;
    }
    if (key == a[mid])
        printf ("\n%d element found at %d posotion.\n ", key, mid + 1);
    if (start > end)
        printf ("\n%d not found in the list\n ", key);
}
/*
[root@localhost ds]# ./a.out
How many element you want to enter in the array : 5
Enter the  element 1 :2
Enter the  element 2 :4
Enter the  element 3 :1
Enter the  element 4 :9
Enter the  element 5 :3
Data after Sorting:
       1       2       3       4       9
Enter the element to be searched  :3
3 element found at 3 posotion.

[root@localhost ds]# ./a.out
How many element you want to enter in the array : 5
Enter the  element 1 :6
Enter the  element 2 :1
Enter the  element 3 :3
Enter the  element 4 :2
Enter the  element 5 :9
Data after Sorting:
       1       2       3       6       9
Enter the element to be searched  :8
8 not found in the list

*/
```

Set B

a) **Read the data from file 'cities.txt' containing names of cities and their STD codes. Accept a name of the city from user and use linear search algorithm to check whether the name is present in the file and output the STD code, otherwise output "city not in the list".**

Cities.txt

```
21 pune
22 mumbai
```

54 hyderabad
76 delhi

```
// Linear search on city
#include<stdio.h>
#include<stdlib.h>
struct cities
{
        char city[20];
        int code;

};
typedef struct cities cit;
int readfile(cit *);
void display(cit *,int);
void linearsearch(cit *,int, char*);
main()
{
  int n;
  char key[30];
        cit a[100];
        n=readfile(a);
        printf("\nStructure data after reading file: ");
        display(a,n);
       printf ("\nEnter the city to be search :");
        scanf ("%s", key);
        linearsearch(a,n,key);
}
int readfile(cit *a)
{
        char fname[20];
        int i=0;
        FILE *fp;
        printf("\nEnter file name to read: ");
        scanf("%s",fname);
        fp=fopen(fname,"r");
        if(fp==NULL)
        {
                printf("\nError in opening file!");
                exit(0);
        }
        while(!feof(fp))
        {
                fscanf(fp,"%d %s",&a[i].code,a[i].city);
                i++;
        }
        return i-1;
}

void display(cit *a,int n)
{
```

```
        int i;
        for(i=0;i<n;i++)
                printf("\n%d %s",a[i].code,a[i].city);
}
void linearsearch(cit *a,int n,char *key)
{
        int i;
        for (i = 0; i < n; i++)
        {
          if (strcmp(a[i].city,key)==0)
                {
                  printf ("\n  %s city found \n Code of %s city is %d.", key,key,a[i].code);
                        break;
                }
        }

        if (i == n)
                printf ("\nCity not found in the list\n");
}
/*
[root@localhost ds]# cc ass1setB1.c
[root@localhost ds]# ./a.out

Enter file name to read: cities.txt

Structure data after reading file:
21 pune
22 mumbai
54 hyderabad
76 delhi
Enter the city to be search :mumbai

  mumbai city found
 Code of mumbai city is 22.
[root@localhost ds]# ./a.out

Enter file name to read: cities.txt

Structure data after reading file:
21 pune
22 mumbai
54 hyderabad
76 delhi
Enter the city to be search :nagpur

City not found in the list

*/
```

b) **Read the data from file 'cities.txt' containing names of cities and their STD codes. Accept a name of the city from user and use sentinel linear search algorithm to check whether the name is present in the file and output the STD code, otherwise output "city not in the list".**

```
//Sentinel Linear search on city
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct cities
{
        char city[20];
        int code;

};
typedef struct cities cit;
int readfile(cit *);
void display(cit *,int);
void sentinelsearch(cit *,int, char*);
main()
{
  int n;
  char key[30];
        cit a[100];
        n=readfile(a);
        printf("\nStructure data after reading file: ");
        display(a,n);
       printf ("\nEnter the city to be search :");
        scanf ("%s", key);
        sentinelsearch(a,n,key);
}
int readfile(cit *a)
{
        char fname[20];
        int i=0;
        FILE *fp;
        printf("\nEnter file name to read: ");
        scanf("%s",fname);
        fp=fopen(fname,"r");
        if(fp==NULL)
        {
                printf("\nError in opening file!");
                exit(0);
        }
        while(!feof(fp))
        {
                fscanf(fp,"%d %s",&a[i].code,a[i].city);
                i++;
        }
        return i-1;
}

void display(cit *a,int n)
```

```
{
        int i;
        for(i=0;i<n;i++)
                printf("\n%d %s",a[i].code,a[i].city);
}
void sentinelsearch(cit *a,int n,char *key)
{
        int i=0;
       strcpy(a[n].city,key);
        while(strcmp(a[i].city,key)!=0)
           i++;
      if(i==n)
         printf("\ncity not found!");
       else
                  printf ("\n  %s city found \n Code of %s city is %d.", key,key,a[i].code);


}

/*
[root@localhost ds]# cc ass1setB2.c
[root@localhost ds]# ./a.out

Enter file name to read: cities.txt

Structure data after reading file:
21 pune
22 mumbai
54 hyderabad
76 delhi
Enter the city to be search :delhi

  delhi city found
 Code of delhi city is 76.
[root@localhost ds]# ./a.out

Enter file name to read: cities.txt

Structure data after reading file:
21 pune
22 mumbai
54 hyderabad
76 delhi
Enter the city to be search :agra

city not found!
*/
```

c) Read the data from file 'sortedcities.txt' containing sorted names of cities and their STD codes. Accept a name of the city from user and use binary search algorithm to check whether the name is present in the file and output the STD code, otherwise output "city not in the list".

Sortedcities.txt

```
21 agra
45 delhi
57 mumbai
34 pune
56 satara
```

```
//Binary search on city
#include<stdio.h>
#include<stdlib.h>
struct cities
{
        char city[20];
        int code;

};
typedef struct cities cit;
int readfile(cit *);
void display(cit *,int);
void binarysearch(cit *,int, int,char *);
main()
{
      int n;
      char key[30];
        cit a[100];
        n=readfile(a);
        printf("\nStructure data after reading file: ");
        display(a,n);
      printf ("\nEnter the city to be search :");
        scanf ("%s", key);
        binarysearch(a,0,n-1,key);
}
int readfile(cit *a)
{
        char fname[20];
        int i=0;
        FILE *fp;
        printf("\nEnter file name to read: ");
        scanf("%s",fname);
        fp=fopen(fname,"r");
        if(fp==NULL)
        {
                printf("\nError in opening file!");
                exit(0);
        }
```

```
        while(!feof(fp))
        {
                fscanf(fp,"%d %s",&a[i].code,a[i].city);
                i++;
        }
        return i-1;
}

void display(cit *a,int n)
{
        int i;
        for(i=0;i<n;i++)
                printf("\n%d %s",a[i].code,a[i].city);
}
void binarysearch(cit *a,int start, int end,char *key)
{
        int mid;
        mid = (start + end) / 2;
        while ((strcmp(key, a[mid].city)!=0) && (start <= end))
        {
          if (strcmp(key, a[mid].city)>0)
                        start = mid + 1;
                else
                        end=mid - 1;
                mid = (start + end) / 2;
        }
        if (strcmp(key, a[mid].city)==0)
          printf ("\n  %s city found \n Code for %s city is %d.\n ", key,key,a[mid].code);
        if (start > end)
                printf ("\nCity not in the list\n ");
}
/*
[root@localhost ds]# cc ass1setB3.c
[root@localhost ds]# ./a.out

Enter file name to read: sortedcities.txt

Structure data after reading file:
21 agra
45 delhi
57 mumbai
34 pune
56 satara
Enter the city to be search :pune

  pune city found
 Code for pune city is 34.
 [root@localhost ds]# ./a.out

Enter file name to read: sortedcities.txt
```

Structure data after reading file:
21 agra
45 delhi
57 mumbai
34 pune
56 satara
Enter the city to be search :hyderabad

City not in the list

*/