

```

/*
Experiment No.           : 01
Filename                 : ArrayADT
Author                   : Mehul Y Khandhadiya
Date                     : 10/07/19
Aim                      : To study an array ADT and to implement various operations on array ADT
Problem Statement :      : Create an array and implement the operations -
                           initialize( ), create( ), isFull( ), isEmpty( ), copy( ),
                           traverse( ), length( ), insert_element( ), merge( ),
                           delete_element( ), sort( ), search( ), display( ),
                           Write a c program to demonstrate an array ADT using defined operations
                           appropriately using a menu- driven program approach.
                           Your program should be able to print the array contents appropriately at any
                           or all instances
                           You must also ensure that no input is acquired.
*/

```

```

/*
/*Pre-processor Directives*/
#include <stdio.h>
#include<stdlib.h>
#define MIN_VAL -99
#define MAX_VAL 5

/* Function declaration */
int initialize(int arr[]);
int create(int arr[],int len);
int length(int arr[]);
void display(int arr[],int len);
void traverse(int arr[],int len);
int insert(int arr[],int len, int key, int value);
int delete(int arr[],int len,int key);
int isEmpty(int arr[]);
int isFull(int arr[]);
void copy(int arr[],int len);
void merge(int arr[],int barr[],int blen,int len);
void sort(int arr[],int len);
int search(int arr[],int len, int key);

/* Main Function */
int main( )
{
    int i=0;
    int arr[MAX_VAL],barr[MAX_VAL];
    int choice,len,key,dkey,value,blen;
    printf("An array adt");
    len= initialize(arr);
    printf("\ncreate an array ?\nEnter 1 for YES or press any other key to EXIT : ");
    scanf("%d",&choice);
    if(choice!=1)
    {
        printf(" Terminating condition\n");
        exit(1);
    }
    len= create(arr,len);
do{

```

```

printf("\nChoose an Array ADT operation\n");
printf("\n 1. length  2. display  3.traverse  4. insert");
printf("\n 5. delete  6. copy  7. merge  8. Check isEmpty");
printf("\n 9. Check isFull 10. Sort 11. Search 0. Exit");
printf("\n Enter your operation code?");
scanf("%d",&choice);
    switch(choice)
    {
        case 0:
            printf("\nTerminating array\n");
            break;
        case 1:
            printf("\n Array length is : %d",length(arr));
            break;
        case 2:
            printf(" Displaying Array \n");
            display(arr,len);
            break;
        case 3:
            printf(" Traversal of Array \n");
            traverse(arr,len);
            break;
        case 4:
            do
            {
                printf("\n Enter index\n");
                scanf("%d",&key);
            }while(key<0 && key>=len);
            printf("\n Enter element\n");
            scanf("%d",&value);
            if(value=-99)
            {
                printf("Terminating value entered");
                break;
            }
            else
            {
                len=insert(arr,len,key,value);
                break;
            }
        case 5:
            printf("\n Enter index\n\n");
            scanf("%d",&dkey);
            len=delete(arr,len,dkey);
            break;
        case 6:
            printf(" Copying of array \n");
            copy(arr,len);
            break;
        case 7:
            printf("\n Enter the elements of second array \n");
            blen=initialize(barr);
            blen=create(barr,blen);
            print("Merging array\n");

```

```

        merge(arr,barr,blen,len);
        break;
case 8:
    if(isEmpty(arr))
        printf("\nArray is Empty\n");
    else
        printf("\nArray is not Empty\n");
    break;
case 9:
    if(isFull(arr))
        printf("\nArray is Full\n");
    else
        printf("\nArray is not Full \n");
    break;
case 10:
    sort(arr,len);
    break;
case 11:
    printf("Enter the key to be searched:");
    scanf("%d",&key);
    printf("\nEnter 1.Linear Search\t 2. Binary Search");
    scanf("%d",&choice);
    if(choice==1)
    {
        linear_search(arr,len,key);
    }
    else
    {
        if(choice==2)
        {
            binary_search(arr,len,key);
        }
    }
    break;
default:
    printf("\n Invalid input ");
    break;

    }
}while(choice!=0);
return 0;
}

/* Function definition */

int initialize(int arr[])
{
    arr[0]=MIN_VAL;
    return 0;
}
int create(int arr[],int len)
{
    int val;
    while(
        {

```

```

        if(len>=MAX_VAL)
        {
            printf("\nArray Bounds Reached\n");
            break;
        }
    else
    {
        printf("Enter a Value (ENTER -99 TO EXIT)");
        scanf("%d",&val);
        arr[len]=val;
        if(val==-99)
            break;

        len++;
    }
}
return len;
}
int length(int arr[])
{
    int len=0;
    while(len<=MAX_VAL )
    {
        if( arr[len]==MIN_VAL)
            break;

        len++;
    }
    return len;
}

void display(int arr[],int len)
{
    int i;
    for(i=0;i<len;i++)
    {
        printf("%d - %d \n",arr[i],&arr[i]);
    }
}

void traverse(int arr[],int len)
{
    int i;
    for(i=0;i<len;i++)
    {
        printf("arr[%d]=%d ",i,arr[i]);
    }
}

int insert(int arr[],int len, int key, int value)
{
    int j;
    for(j=len;j>=key;j--)
    {
        arr[j]=arr[j-1];
    }
    len++;
    arr[key]=value;
}

```

```

        display(arr,len);
        arr[len]=MIN_VAL;
    return len;
}
int delete(int arr[],int len,int key)
{
    int x;
    for(x=key;x<len;x++)
    {
        arr[x]=arr[x+1];
    }
    len--;
    display(arr,len);

    return len;
}
int isEmpty(int arr[])
{
    if(length(arr)==0)
        return 1;
    else
        return 0;
}
int isFull(int arr[])
{
    if(length(arr)==MAX_VAL)
        return 1;
    else
        return 0;
}
void copy(int arr[],int len)
{
    int i,arr1[len];
    for(i=0;i<=len;i++)
        arr1[i]=arr[i];
    display(arr1,len);
}
void merge(int arr[],int barr[],int blen,int len)
{
    int i,j,alen,mlen;
    alen=len;
    mlen=alen+blen;
    int merged[mlen];
    for(i =0;i<=alen; i++)
        merged[i] = arr[i];
    for(i = 0, j = alen; j < mlen && i < blen; i++, j++)
        merged[j] = barr[i];
    display(merged,mlen);
}
void sort(int arr[],int len)
{
    int i,j,choice,temp;
    printf("\nsorting of elements.");
    printf("\n1.ascending\t2.descending:");
    scanf("%d",&choice);

```

```

for(i=0;i<len;i++)
{
    for(j=i+1;j<len;j++)
    {
        if(choice==1)
        {
            if(arr[i]>arr[j])
            {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
        if(choice==2)
        {
            if(arr[i]<arr[j])
            {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
}
printf("\nsorting done.");
}
int linear_search(int arr[],int len,int key)
{
    int i;
    for(i=0;i<len;i++)
    {
        if(key==arr[i])
        {
            printf("\nKey found.Search successfull.\n");
            return 1;
        }
    }
    printf("Key not found.\n");
    return 0;
}
int binary_search(int arr[],len,key)
{
    int l,u,mid;
    l=0,u=len;
    while(l<=u)
    {
        mid=(l+u)/2;
        if(arr[mid]==num)
        {
            printf("Key is present");
        }
        else
        {
            if(key>mid)

```

```

        {
            l=(mid+1);
        }
        else
        {
            u=(mid-1);
        }
    }
}
return 1;
}

```

/*

EXECUTION TRAIL

An array adt

create an array ?

Enter 1 for YES or press any other key to EXIT : 1

Enter a Value (ENTER -99 TO EXIT)22

Enter a Value (ENTER -99 TO EXIT)66

Enter a Value (ENTER -99 TO EXIT)33

Enter a Value (ENTER -99 TO EXIT)-99

Choose an Array ADT operation

1. length 2. display 3.traverse 4. insert

5. delete 6. copy 7. merge

8. Check isEmpty 9. Check isFull

10. Sort 11.Search

0. Exit

Enter your operation code?1

Array length is : 3

Choose an Array ADT operation

1. length 2. display 3.traverse 4. insert

5. delete 6. copy 7. merge

8. Check isEmpty 9. Check isFull

10. Sort 11. Search

0. Exit

Enter your operation code?2

Displaying Array

22 - 1959849472

66 - 1959849476

33 - 1959849480

Choose an Array ADT operation

1. length 2. display 3.traverse 4. insert

5. delete 6. copy 7. merge

8. Check isEmpty 9. Check isFull
10. Sort 11. Search
0. Exit
Enter your operation code?3

Traversing Array
arr[0]=22 arr[1]=66 arr[2]=33

Choose an Array ADT operation

1. length 2. display 3.traverse 4. insert
5. delete 6. copy 7. merge
8. Check isEmpty 9. Check isFull
10. Sort 11. Search
0. Exit
Enter your operation code?4

Enter index
2

Enter element
44
22 - 1959849472
66 - 1959849476
44 - 1959849480
33 - 1959849484

Choose an Array ADT operation

1. length 2. display 3.traverse 4. insert
5. delete 6. copy 7. merge
8. Check isEmpty 9. Check isFull
10. Sort 11. Search
0. Exit
Enter your operation code?5

Enter index

2
22 - 1959849472
66 - 1959849476
33 - 1959849480

Choose an Array ADT operation

1. length 2. display 3.traverse 4. insert
5. delete 6. copy 7. merge
8. Check isEmpty 9. Check isFull
10. Sort 11. Search
0. Exit
Enter your operation code?6

Copying of array

22 - 1959849184

66 - 1959849188

33 - 1959849192

Choose an Array ADT operation

1. length 2. display 3. traverse 4. insert

5. delete 6. copy 7. merge

8. Check isEmpty 9. Check isFull

10. Sort 11. Search

0. Exit

Enter your operation code?7

Enter the elements of second array

Enter a Value (ENTER -99 TO EXIT)32

Enter a Value (ENTER -99 TO EXIT)65

Enter a Value (ENTER -99 TO EXIT)-99

Merging array

22 - 1959849136

66 - 1959849140

33 - 1959849144

32 - 1959849148

65 - 1959849152

Choose an Array ADT operation

1. length 2. display 3. traverse 4. insert

5. delete 6. copy 7. merge

8. Check isEmpty 9. Check isFull

10. Sort 11. Search

0. Exit

Enter your operation code?8

Array is not Empty

Choose an Array ADT operation

1. length 2. display 3. traverse 4. insert

5. delete 6. copy 7. merge

8. Check isEmpty 9. Check isFull

10. Sort 11. Search

0. Exit

Enter your operation code?2

Displaying Array

22 - 1959849472

66 - 1959849476

33 - 1959849480

Choose an Array ADT operation

- 1. length 2. display 3. traverse 4. insert
- 5. delete 6. copy 7. merge
- 8. Check isEmpty 9. Check isFull
- 10. Sort 11. Search
- 0. Exit

Enter your operation code?9

Array is not Full

Choose an Array ADT operation

- 1. length 2. display 3. traverse 4. insert
- 5. delete 6. copy 7. merge
- 8. Check isEmpty 9. Check isFull
- 10. Sort 11. Search
- 0. Exit

Enter your operation code?10

sorting of elements.

1. ascending 2. descending:1

sorting done.

Choose an Array ADT operation

- 1. length 2. display 3. traverse 4. insert
- 5. delete 6. copy 7. merge
- 8. Check isEmpty 9. Check isFull
- 10. Sort 11. Search
- 0. Exit

Enter your operation code?2

Displaying Array

22 - 1959849472

33 - 1959849476

66 - 1959849480

Choose an Array ADT operation

- 1. length 2. display 3. traverse 4. insert
- 5. delete 6. copy 7. merge
- 8. Check isEmpty 9. Check isFull
- 10. Sort 11. Search
- 0. Exit

Enter your operation code?11

Enter the key to be searched:22

Enter 1. Linear Search 2. Binary Search

Key found.Search successfull.

Choose an Array ADT operation

- 1. length 2. display 3.traverse 4. insert
- 5. delete 6. copy 7. merge
- 8. Check isEmpty 9. Check isFull
- 10. Sort 11. Search
- 0. Exit

Enter your operation code?0

Terminating array

*/k isEmpty 9. Check isFull

- 10. Sort 11. Search

- 0. Exit

Enter your operation code?10

sorting started...

1.ascending 2.descending:1

sorting done.

Choose an Array ADT operation

- 1. length 2. display 3.traverse 4. insert
- 5. delete 6. copy 7. merge
- 8. Check isEmpty 9. Check isFull
- 10. Sort 11. Search
- 0. Exit

Enter your operation code?2

Displaying Array

22 - 1959849472

33 - 1959849476

66 - 1959849480