

# ASSIGNMENT 1 SOLUTIONS

1) Develop a Menu driven program to demonstrate the following operations of Arrays

——MENU——

1. CREATE
2. DISPLAY
3. INSERT
4. DELETE
5. SEARCH (Linear)
6. EXIT

```
1  #include<stdio.h>
2  int main()
3  {
4      int a[6], i, pos, p, ele, n, ser;
5      printf("Enter your choice:\n1. CREATE \n2.DISPLAY \n3.INSERT \n4.DELETE \n5.SEARCH(Linear) \n6.EXIT");
6      scanf("%d", &n);
7      while(n!=6)
8      {
9          switch(n)
10         {
11             case 1:
12             {
13                 printf("Enter the elements for your array");
14                 for(i=0;i<5;i++)
15                 {
16                     scanf("%d", &a[i]);
17                 }
18                 break;
19             }
20             case 2:
21             {
22                 printf("Here is your created array");
23                 for(i=0;i<5;i++)
24                 {
25                     printf("\n%d", a[i]);
26                 }
27                 break;
28             }
29             case 3:
30             {
31
32
```

```

31 |
32 |
33 |         printf("Enter the position where you want to insert the element");
34 |         scanf("%d", &p);
35 |         printf("Enter the element");
36 |         scanf("%d", &ele);
37 |         for(i=5;i>p-1;i--)
38 |         {
39 |             a[i]=a[i-1];
40 |         }
41 |         a[p-1]=ele;
42 |         printf("The array is now as follows");
43 |         for(i=0;i<6;i++)
44 |         {
45 |             printf("%d", a[i]);
46 |         }
47 |         break;
48 |     }
49 |     case 4:
50 |     {
51 |         printf("Enter the position of the element which you want to delete");
52 |         scanf("%d", &pos);
53 |         for(i=pos-1;i<5;i++)
54 |         {
55 |             a[i]=a[i+1];
56 |         }
57 |         printf("The array is now as follows");
58 |         for(i=0;i<4;i++)
59 |         {
60 |             printf("%d", a[i]);
61 |         }
62 |         break;
63 |
64 |     }
65 |     case 5:
66 |     {
67 |         printf("Enter the element you want to search");
68 |         scanf("%d", &ser);
69 |         for(i=0;i<5;i++)
70 |         {
71 |             if(a[i]==ser)
72 |             {
73 |                 printf("Element is present at position %d", i+1);
74 |             }
75 |         }
76 |         break;
77 |     }
78 |     default:
79 |         printf("Error");
80 |
81 | }
82 | printf("Enter your choice:\n1. CREATE \n2.DISPLAY \n3.INSERT \n4.DELETE \n5.SEARCH(Linear) \n6.EXIT");
83 | scanf("%d", &n);
84 | }
85 | return 0;
86 |

```

2) Design the logic to remove the duplicate elements from an Array and after the deletion, the array should contain the unique elements

```

9  #include <iostream>
10
11  using namespace std;
12
13  int main()
14  {
15      int n;
16      cin>>n;
17      int arr[n];
18      for(int i=0;i<n;i++){
19          cin>>arr[i];
20      }
21      cout<<"Array before: ";
22      for(int i=0;i<n;i++){
23          cout<<arr[i]<<" ";
24      }
25      cout<<endl;
26      for(int i=0;i<n;i++){
27          for(int j=i+1;j<n;j++){
28              if(arr[j]==arr[i]){
29                  for(int k=j;k<n-1;k++){
30                      arr[k]=arr[k+1];
31                  }
32                  n--;
33                  j--;
34              }
35          }
36      }
37      cout<<"Array after: ";
38      for(int i=0;i<n;i++){
39          cout<<arr[i]<<" ";
40      }
41      return 0;
42  }
43  }

```

3) Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order. Code the Bubble sort with the following elements:

```

9  #include <iostream>
10
11 using namespace std;
12
13 int main()
14 {
15     int n;
16     cout<<"Enter Array Size: ";
17     cin>>n;
18     int arr[n];
19     cout<<"Enter array elements: ";
20     for(int i=0;i<n;i++){
21         cin>>arr[i];
22         cout<<" ";
23     }
24     /*Bubble Sort*/
25     for(int i=0;i<n;i++){
26         for(int j=0;j<n-1;j++){
27             if(arr[j]>arr[j+1]){
28                 int temp=arr[j];
29                 arr[j]=arr[j+1];
30                 arr[j+1]=temp;
31             }
32         }
33     }
34     cout<<"Sorted array is:- \n";
35     for(int i=0;i<n;i++){
36         cout<<arr[i]<<" ";
37     }
38     int x,mid,low, high,flag;
39     cout<<"Enter the element to search: ";
40     cin>>x;
41     low=0;
42     high=n-1;

```

```

43     for(int i=low;i<high;i++){
44         mid=(low+high)/2;
45         if(x==mid){
46             cout<<"Element found";
47             flag=1;
48             break;
49         }
50         else if(x>mid){
51             low=mid+1;
52         }
53         else{
54             high=mid+1;
55         }
56     }
57     if(flag!=1){
58         cout<<"Element Not Found";
59     }
60
61     return 0;
62 }
63

```

4) Develop a program to find the minimum and maximum elements in an array.

```
1  #include<stdio.h>
2  int main()
3  {
4      int a[7], i, max, min;
5      printf("Enter the elements");
6      for(i=0;i<7;i++)
7      {
8          scanf("%d", &a[i]);
9      }
10     max=a[0]; min=a[0];
11     for(i=0;i<7;i++)
12     {
13         if (a[i]>max)
14         {
15             max=a[i];
16         }
17         if (a[i]<min)
18         {
19             min = a[i];
20         }
21     }
22     printf(" Max = %d; Min = %d", max, min);
23     return 0;
24 }
25
```

5) Develop the following programs

a) Printing the diagonal, upper and lower diagonal elements of a matrix.

```

#include<stdio.h>
int main()
{
    int i, j, a[3][3];
    printf("Enter the elements for your 5x5 matrix row wise");
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            scanf("%d", &a[i][j]);
        }
    }
    printf("\n");
    printf("Diagonal elements\n");
    public int __cdecl printf (const char * __restrict__ _Format, ...)
    {
        for(j=0;j<3;j++)
        {
            if(i==j)
            {
                printf("%d", a[i][j]);
            }
            else
            printf(" ");
        }
        printf("\n");
    }
    printf("\n");
}

```

```

printf("\n");
printf("Upper diagonal\n");
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        if(i<j||i==j)
        {
            printf("%d", a[i][j]);
        }
        else
            printf(" ");
    }
    printf("\n");
}
printf("\n");
printf("Lower diagonal\n");
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        if(i>j||i==j)
        {
            printf("%d", a[i][j]);
        }
        else
            printf(" ");
    }
    printf("\n");
}
return 0;
}

```

**b)Find the row-wise and column-wise sum of all the elements that present in a two-dimensional array.**



```

1  #include<stdio.h>
2  int main()
3  {
4      int a[3][3], i, j, sum;
5      for(i=0;i<3; i++)
6      {
7          for( j=0;j<3;j++)
8          {
9              scanf("%d", &a[i][j]);
10             }
11         }
12         for(i=0;i<3;i++)
13         {
14             sum=0;
15             for(j=0;j<3; j++)
16             {
17                 sum=sum+a[i][j];
18             }
19             printf("\nSum of row %d is %d", i+1, sum);
20         }
21         for(j=0;j<3;j++)
22         {
23             sum=0;
24             for(i=0;i<3; i++)
25             {
26                 sum=sum+a[i][j];
27             }
28             printf("\nSum of column %d is %d", j+1, sum);
29         }
30     }
31     public int __cdecl printf (const char * __restrict __Format, ...)
32 }

```