

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX_SIZE 100
5
6  void traverseArray(int arr[], int size);
7
8  /*size means size is being passed as call by reference because it will be changed after
9  every insertion and deletion operation
10 */
11 void insertInArray(int arr[], int *size);
12 void deleteInArray(int arr[], int *size);
13
14 void updateArray(int arr[], int size);
15 void sortArray(int arr[], int size);
16 void searchArray(int arr[], int size);
17
18 int main()
19 {
20     int arr[MAX_SIZE];
21     int size = 0;
22     int choice;
23
24     do
25     {
26         printf("\nArray Operations\n");
27         printf("1. Traverse\n");
28         printf("2. Insert\n");
29         printf("3. Delete\n");
30         printf("4. Update\n");
31         printf("5. Sort\n");
32         printf("6. Search\n");
33         printf("7. Exit\n");
34
35         printf("\nEnter your choice: ");
36         scanf("%d", &choice);
37
38         switch(choice)
39         {
40             case 1:
41                 traverseArray(arr, size);
42                 break;
43             case 2:
44                 insertInArray(arr, &size);
45                 break;
46             case 3:
47                 deleteInArray(arr, &size);
48                 break;
49             case 4:
50                 updateArray(arr, size);
51                 break;
52             case 5:
53                 sortArray(arr, size);
54                 break;
55             case 6:
56                 searchArray(arr, size);
57                 break;
58             case 7:
59                 exit(0);
60             default:
61                 printf("\nInvalid choice! Try again.\n");
62         }
63     } while (choice != 7);
64
65     return 0;
66 }
67
68 void traverseArray(int arr[], int size)

```

```

69 {
70     int i;
71     if (size == 0)
72     {
73         printf("\nThe array is empty.\n");
74         return;
75     }
76
77     printf("\nThe elements of the array are: ");
78     for ( i = 0; i < size; i++)
79     {
80         printf("%d ", arr[i]);
81     }
82     printf("\n");
83 }
84
85 void insertInArray(int arr[], int *size)
86 {
87     int i, index, value;
88     if (*size == MAX_SIZE)
89     {
90         printf("\nThe array is full. Cannot insert any more elements.\n");
91         return;
92     }
93
94     printf("\nEnter the index where you want to insert the element: ");
95     scanf("%d", &index);
96
97     if (index < 0 || index > *size)
98     {
99         printf("\nInvalid index. Try again.\n");
100         return;
101     }
102
103     printf("\nEnter the value of the element: ");
104     scanf("%d", &value);
105
106     for ( i = *size; i > index; i--)
107     {
108         arr[i] = arr[i-1];
109     }
110
111     arr[index] = value;
112     (*size)++; //array size will be increased after every successful insertion
113
114     printf("\nElement inserted successfully.\n");
115 }
116
117 void deleteInArray(int arr[], int *size)
118 {
119     int i, index;
120     if (*size == 0)
121     {
122         printf("\nThe array is empty. Cannot delete any elements.\n");
123         return;
124     }
125
126
127     printf("\nEnter the index of the element you want to delete: ");
128     scanf("%d", &index);
129
130     if (index < 0 || index >= *size)
131     {
132         printf("\nInvalid index. Cannot delete the element.\n");
133         return;
134     }
135
136     for ( i = index; i < *size-1; i++)
137     {

```

```

138         arr[i] = arr[i+1];
139     }
140
141     (*size)--; //array size will be decreased after every successful deletion
142
143     printf("\nElement deleted successfully.\n");
144 }
145
146 void updateArray(int arr[], int size)
147 {
148     int index, value;
149     printf("\nEnter the index of the element you want to update: ");
150     scanf("%d", &index);
151     if (index < 0 || index >= size)
152     {
153         printf("\nInvalid index. Cannot update the element.\n");
154         return;
155     }
156     printf("\nEnter the updated value: ");
157     scanf("%d", &value);
158     arr[index] = value;
159
160     printf("\nElement updated successfully.\n");
161 }
162
163 void searchArray(int arr[], int size)
164 {
165     int i, target;
166     printf("\nEnter the value to be searched:");
167     scanf("%d",&target);
168     for (i = 0; i < size; i++)
169     {
170         if (arr[i] == target)
171         {
172             printf("%d found at index %d\n", target,i);
173             return;
174         }
175     }
176     printf("%d not found in the array\n",target);
177 }
178 void sortArray(int arr[], int size) //Bubble Sort
179 {
180     int i,j,temp;
181
182     for ( i = 0; i < size-1; i++)
183     {
184         for ( j = 0; j < size-i-1; j++)
185         {
186             if (arr[j] > arr[j+1])
187             {
188                 temp = arr[j];
189                 arr[j] = arr[j+1];
190                 arr[j+1] = temp;
191             }
192         }
193     }
194
195     printf("\nArray sorted in ascending order through Bubble Sort.\n");
196     traverseArray(arr,size);
197 }
198

```