

Bhavleen Kaur
3EIC-2
102155010

LAB ASSIGNMENT 2

ARRAYS AND POINTERS

1. Write a program to check whether a given number is present in an array or not (Linear search).

```
1  #include <stdio.h>
2
3- int main() {
4      int key;
5      printf("Enter the key to be searched: ");
6      scanf("%d", &key);
7      printf("The key to be searched is %d\n", key);
8
9      int n;
10     printf("Enter the size of array: ");
11     scanf("%d", &n);
12     printf("The size of array is %d\n", n);
13
14     int arr[n];
15
16     printf("Enter the elements of array: ");
17-    for (int i = 0; i < n; i++) {
18        scanf("%d", &arr[i]);
19    }
20
21     int found = 0;
22
23-    for (int i = 0; i < n; i++) {
24-        if (arr[i] == key) {
25            printf("Key Found!!\n");
26            found = 1;
27            break;
28        }
29    }
30
31-    if (!found) {
32        printf("Key not found.\n");
33    }
34
35     return 0;
36 }
```

```

/tmp/qQgKULYEad.o
Enter the key to be searched: 52
The key to be searched is 52
Enter the size of array: 5
The size of array is 5
Enter the elements of array: 56
-52
85
-85
52
Key Found!!

```

2. Write a program to get second maximum and second minimum elements in an array.

```

1  #include <stdio.h>
2
3  void secondMaxMin(int arr[], int n) {
4      int max = arr[0], secondMax = arr[0];
5      int min = arr[0], secondMin = arr[0];
6
7      for (int i = 1; i < n; i++) {
8          if (arr[i] > max) {
9              secondMax = max;
10             max = arr[i];
11         } else if (arr[i] > secondMax && arr[i] != max) {
12             secondMax = arr[i];
13         }
14
15         if (arr[i] < min) {
16             secondMin = min;
17             min = arr[i];
18         } else if (arr[i] < secondMin && arr[i] != min) {
19             secondMin = arr[i];
20         }
21     }
22
23     printf("Second Maximum: %d\n", secondMax);
24     printf("Second Minimum: %d\n", secondMin);
25 }
26
27 int main() {
28     int n;
29     printf("Enter the number of elements in the array: ");
30     scanf("%d", &n);
31
32     int arr[n];
33     printf("Enter the elements of the array:\n");
34     for (int i = 0; i < n; i++) {
35         scanf("%d", &arr[i]);
36     }
37
38     secondMaxMin(arr, n);
39
40     return 0;
41 }
/tmp/qQgKULYEad.o
Enter the number of elements in the array: 5
Enter the elements of the array:
-85
85
-65
65
45
Second Maximum: 65
Second Minimum: -85

```

3. Write a program to perform insertion (any location), deletion (any location) and traversal in an array.

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int a[100];
6      int n;
7
8      printf("Enter the size of an array\n");
9      scanf("%d",&n);
10
11     printf("Enter the elements of array: ");
12     for(int i=0;i<n;i++)
13     {
14         scanf("%d",&a[i]);
15     }
16     printf("List before Insertion: ");
17     for(int i=0;i<n;i++)
18     {
19         printf("%d ",a[i]);
20     }
21
22     int element;
23     printf("\nEnter an element to insert\n");
24     scanf("%d",&element);
25     printf("Enter the position to insert an element %d\n",element);
26     int loc;
27     scanf("%d",&loc);
28     loc--;
29     for(int i=n-1;i>=loc;i--)
30     {
31         a[i+1]=a[i];
32     }
33     a[loc]=element;
34     printf("\nList after Insertion: ");
35     for(int i=0;i<n+1;i++)
36     {
37         printf("%d ",a[i]);
38     }
39     printf("\nEnter an element to delete\n");
40     scanf("%d",&n);
41     int i=0;
42     for(int i=0;i<n;i++)
43     {
44         if(a[i]==n)
45         {
46             for(int j=i;j<(n-1);j++)
47             {
48                 a[j]=a[j+1];
49             }
50             break;
51         }
52     }
53     printf("List after deletion\n");
54     for(i=0;i<(n-1);i++)
55     {
56         printf("%d ",a[i]);
57     }
58     return 0;
```

```

/tmp/qQgKULYEad.o
Enter the size of an array
5
Enter the elements of array: -58
58
68
-68
23
List before Insertion: -58 58 8 -68 3
Enter an element to insert 56

Enter the position to insert an element 6
4
List after Insertion: -58 58 8 6 -68 3

```

4. Write a menu driven program to perform addition, multiplication and subtraction of 2 arrays.

```

1  #include<stdio.h>
2- int SumArrays(int arr1[], int arr2[], int result[], int n){
3-     for(int i=0;i<n;i++){
4         result[i] = arr1[i]+arr2[i];
5     }
6 }
7
8- int DiffArrays(int arr1[], int arr2[], int result[], int n){
9-     for(int i=0;i<n;i++){
10        result[i] = arr1[i]-arr2[i];
11    }
12 }
13
14- int ProdArrays(int arr1[], int arr2[], int result[], int n){
15-     for(int i=0;i<n;i++){
16        result[i] = arr1[i]*arr2[i];
17    }
18 }
19- void displayArray(int arr[], int n) {
20     printf("[ ");
21-     for (int i = 0; i < n; i++) {
22         printf("%d ", arr[i]);
23     }
24     printf("]\n");
25 }
26- int main(){
27     int n;
28     int result[n];
29
30     printf("Enter a number: ");
31     scanf("%d", &n);
32     int arr1[n];
33     int arr2[n];
34-     for (int i =0;i<n;i++){
35         scanf("%d", &arr1[i]);
36     }
37-     for (int i =0;i<n;i++){
38         scanf("%d", &arr2[i]);
39     }
40

```

```
41     int choice;
42     printf("Enter the choice: ");
43     scanf("%d", &choice);
44     switch (choice)
45     {
46     case 1:
47         SumArrays(arr1, arr2,result, n);
48         displayArray(result, n);
49         break;
50     case 2:
51         DiffArrays(arr1, arr2,result, n);
52         displayArray(result, n);
53         break;
54     case 3:
55         ProdArrays(arr1, arr2,result, n);
56         displayArray(result, n);
57         break;
58
59     default:
60         break;
61     }
62     return 0;
```

5. Write a program to perform sorting while merging (Merge two sorted arrays into one sorted array).

```
1  #include<stdio.h>
2
3- void MergeArrays(int arr1[], int n1, int arr2[], int n2, int result[]){
4      int i=0,j=0,k=0;
5-      while(i<n1 && j<n2){
6-          if (arr1[i]<arr2[j]){
7              result[k++]=arr1[i++];
8          }
9-          else{
10             result[k++]=arr2[j++];
11         }
12     }
13     while (i<n1){
14         result[k++]=arr1[i++];
15     }
16     while (j<n2){
17         result[k++]=arr2[j++];
18     }
19 }
20
21- void displayArray(int result[], int n1, int n2) {
22     printf("[ ");
23-     for (int i = 0; i < n1+n2; i++) {
24         printf("%d ", result[i]);
25     }
26     printf("]\n");
27 }
28
29- int main(){
30     int n1;
31     printf("Enter the size of first array: ");
32     scanf("%d",&n1);
33     int n2;
34     printf("Enter the size of second array: ");
35     scanf("%d",&n2);
36     int arr1[n1], arr2[n2];
37     printf("Enter the elements of first array: ");
38-     for (int i=0;i<n1;i++){
39         scanf("%d", &arr1[i]);
40     }
41     printf("Enter the elements of second array: ");
42-     for (int i=0;i<n2;i++){
43         scanf("%d", &arr2[i]);
44     }
45     int result[n1+n2];
46     MergeArrays(arr1,n1,arr2,n2,result);
47     printf("Sorted and merged array is: \n");
48     displayArray(result, n1,n2);
49     return 0;
50 }
```

```

/tmp/qQgKULYEad.o
Enter the size of first array: 3
Enter the size of second array: 3
Enter the elements of first array: 4
5
6
Enter the elements of second array: 7
8
9
Sorted and merged array is:
[ 4 5 6 7 8 9 ]

```

6. Write the above programs (1,2, and 3) using functions and call by address only.

a Write a program to check whether a given number is present in an array or not (Linear search).

```

1  #include <stdio.h>
2  int linearSearch(int* arr, int size, int key)
3  {
4      for (int i = 0; i < size; i++) {
5          if (arr[i] == key) {
6              return i;
7          }
8      }
9      return -1;
10 }
11 int main()
12 {
13     int arr[10] = { 3, 4, 1, 7, 5, 8, 11, 42, 3, 13 };
14     int size = sizeof(arr) / sizeof(arr[0]);
15     int key = 4;
16     int index = linearSearch(arr, size, key);
17     if (index == -1) {
18         printf("The element is not present in the arr.");
19     }
20     else {
21         printf("The element is present at arr[%d].", index);
22     }
23
24     return 0;
25 }
/tmp/qQgKULYEad.o
The element is present at arr[1].

```

b Write a program to get second maximum and second minimum elements in an array.

```
1  #include <stdio.h>
2
3  int secondSmallest(int a[],int n)
4+ {
5      int min = a[0];
6      int secondMin = a[1] ;
7      for(int i = 0; i < n; i++)    //First-time Array Traversal
8+ {
9          if(a[i] < min)
10+ {
11              secondMin = min;
12              min = a[i];
13          }
14      }
15      for(int i = 0; i < n; i++)    //Second-time Array Traversal
16+ {
17          if(a[i] < secondMin && a[i] != min)
18+ {
19              secondMin = a[i];
20          }
21      }
22      return secondMin;            //Return the second smallest element
23 }
24 int secondLargest(int a[],int n)
25+ {
26     int max = a[0];
27     int secondMax = a[1] ;
28     for(int i = 0; i < n; i++)    //First-time Array Traversal
29+ {
30         if(a[i] > max)
31+ {
32             secondMax = max;
33             max = a[i];
34         }
35     }
36     for(int i = 0; i < n; i++)    //Second-time Array Traversal
37+ {
38         if(a[i] > secondMax && a[i] != max)
39+ {
40             secondMax = a[i];
```



```

41     }
42 }
43 return secondMax;           //Return the second largest element
44 }
45 int main()
46 {
47     int n;
48     printf("Enter the number of elements:");
49     scanf("%d",&n);
50     printf("Enter the array elements :");
51     int a[n];                //Array Declaration
52     for(int i=0;i<n;i++)
53     {
54         scanf("%d",&a[i]);
55     }
56     if(n<2)
57     {
58         printf("Invalid Input");
59     }
60     else
61     {
62         int sS=secondSmallest(a,n);
63         printf("The second smallest element is %d",sS);
64         printf("\n");
65         int sL=secondLargest(a,n);
66         printf("The second largest element is %d",sL);
67     }
68     return 0;
69 }

```

/tmp/qQgKULYEad.o

Enter the number of elements:5

Enter the array elements :56

56

65

65

65

The second smallest element is 56

The second largest element is 56

- c Write a program to perform insertion (any location), deletion (any location) and traversal in an array.

```
1  #include <stdio.h>
2- int findElement(int arr[], int n, int key){
3-     for (int i=0; i<n; i++){
4-         if(arr[i]==key){
5-             return i;
6-         }
7-     }
8-     return -1;
9- }
10 int main()
11- {
12     int arr[] = { 12, 34, 10, 6, 40 };
13     int n = sizeof(arr) / sizeof(arr[0]);
14     int key = 40;
15     int position = findElement(arr, n, key);
16     if (position == -1)
17         printf("Element not found");
18     else
19         printf("Element Found at Position: %d",
20             position + 1);
21     return 0;
22 }
23
```