



NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY
Department of Computer Science Engineering

LAB PRACTICAL FILE

Data Structures

Professor
Prof. R.S. Rao

Student
Kushagra Lakhwani

Academic Session 2021-2022

Contents

1	Mean and Median	3
1.1	Objective	3
1.2	Algorithm	3
1.3	Code	4
1.4	Output	5
2	Array Insertion/Deletion	6
2.1	Objective	6
2.2	Algorithm	6
2.2.1	Deletion	6
2.2.2	Insertion	6
2.3	Code	6
2.4	Output	7
3	Search inside Array	8
3.1	Objective	8
3.2	Algorithm	8
3.3	Code	8
3.4	Output	9

1 Mean and Median

1.1 Objective

Write a program to find the mean and the median of the numbers stored in an array.

1.2 Algorithm

```
1 Start
2 Step 1 -> declare function to calculate mean
3     double mean(int arr[], int size)
4     declare int sum = 0
5     Loop For int i = 0 and i < size and i++
6     Set sum += arr[i]
7 End
8     return (double)sum/(double)size
9
10 Step 2 -> declare function to calculate median
11     double median(int arr[], int size)
12     call sort(arr, arr+size)
13     IF (size % 2 != 0)
14     return (double)arr[size/2]
15 End
16     return (double)(arr[(size-1)/2] + arr[size/2])/2.0
17
18 Step 3 -> In main()
19     Declare int arr[] = {3,5,2,1,7,8}
20     Declare int size = sizeof(arr)/sizeof(arr[0])
21     Call mean(arr, size)
22     Call median(arr, size)
23 Stop
```

1.3 Code

```
#include <stdio.h>
int main(int argc, char const *argv[])
{
    int arr[50], n;
    printf("Enter the size of the arr:\n");
    scanf("%d", &n);
    printf("enter the elements: \n");
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    printf("Your array is:\n");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
    printf("your sorted array here: \n");
    int i, j, a;
    for (i = 0; i < n; ++i)
        for (j = i + 1; j < n; ++j)
            if (arr[i] > arr[j])
            {
                a = arr[i];
                arr[i] = arr[j];
                arr[j] = a;
            }
    printf("Your array is:\n");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");

    int sum = 0;
    for (int i = 0; i < n; i++)
    {
        int a = arr[i];
        sum += a;
    }
    int mean = sum / n;
    int median = arr[n / 2];

    printf("\n");
    printf("your mean is given as: %d\n", mean);
    printf("your median is given as: %d\n", median);
}
```

1.4 Output

```
Enter the size of the arr:
5

enter the elements:
21
23
44
55
11

Your array is:
21 23 44 55 11
your sorted array here:
Your array is:
11 21 23 44 55

your mean is given as: 30
your median is given as: 23
```

2 Array Insertion/Deletion

2.1 Objective

Write a Program to insert and delete an element from an array.

2.2 Algorithm

2.2.1 Deletion

```
1 Start
2 Set J = K
3 Repeat steps 4 and 5 while J < N
4     Set LA[J] = LA[J + 1]
5     Set J = J+1
6 Set N = N-1
7 Stop
```

2.2.2 Insertion

```
1 Begin
2 IF N = MAX, return
3 ELSE
4 N = N + 1
5 SEEK Location index
6 For All Elements from A[index] to A[N]
7     Move to next adjacent location
8     A[index] = New_Element
9 End
```

2.3 Code

```
#include <stdio.h>
int main(int argc, char const *argv[])
{
    int arr[100];
    int i, item, pos, size = 7;
    printf("Enter 7 elements: ");
    // reading array
    for (i = 0; i < size; i++)
        scanf("%d", &arr[i]);
    // print the original array
    printf("Array before insertion: ");
    for (i = 0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
```

```

    // read element to be inserted
    printf("Enter the element to be inserted: ");
    scanf("%d", &item);
    // read position at which element is to be inserted
    printf("Enter the position at which the element is to be inserted: ");
    scanf("%d", &pos);
    // increase the size
    size++;
    // shift elements forward
    for (i = size - 1; i >= pos; i--)
        arr[i] = arr[i - 1];
    // insert item at position
    arr[pos - 1] = item;
    // print the updated array
    printf("Array after insertion: ");
    for (i = 0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
    return 0;
}

```

2.4 Output

```

Enter 7 elements: 2 4 6 5 1 5 7
Array before insertion: 2 4 6 5 1 5 7
Enter the element to be inserted: 3
Enter the position at which the element is to be inserted: 2
Array after insertion: 2 3 4 6 5 1 5 7

```

3 Search inside Array

3.1 Objective

Write a program to search for a number in array.

3.2 Algorithm

```
1 Start
2   [Initialize counter variable. ] Set i = 0
3   Repeat Step 04 and 05 for i = 0 to i < n
4   if a[i] = x, then jump to step 07
5   [Increase counter. ] Set i = i + 1
6   [End of step 03 loop. ]
7   Print x found at i + 1 position and go to step 09
8   Print x not found (if a[i] != x, after all the iteration of the
9   above for loop. )
10 Stop
```

3.3 Code

```
#include <stdio.h>
int main(int argc, char const *argv[])
{
    int arr[10], Size, i, Search, Flag;
    printf("\n Please Enter the size of an array : ");
    scanf("%d", &Size);
    printf("\n Please Enter %d elements of an array: \n", Size);
    for (i = 0; i < Size; i++)
        scanf("%d", &arr[i]);
    printf("\n Please Enter the Search Element : ");
    scanf("%d", &Search);
    Flag = 0;
    for (i = 0; i < Size; i++)
        if (arr[i] == Search)
        {
            Flag = 1;
            break;
        }
    if (Flag == 1)
        printf("\n We found the Search Element %d at Position %d ", Search, i + 1);
    else
        printf("\n Sorry!! We haven't found the the Search Element %d ", Search);
}
```



```
    return 0;  
}
```

3.4 Output

```
Please Enter the size of an array : 5  
  
Please Enter 5 elements of an array:  
1 2 3 4 5  
  
Please Enter the Search Element : 2  
  
We found the Search Element 2 at Position 2
```