Subject: PRF192- PFC
Workshop 06

Objectives: Managing arrays

Program 1:

Write a program that accepts 10 integers from the user and prints the sum of these 10 numbers on the screen.

For example, if you enter the following numbers:

```
2 7 6 3 4 8 9 7 6 10
```

When the code is compiled and executed, it produces the following result:

```
62
```

PROGRAM 2:

Write a program that accepts an array `arr` of `n` integers from the user and prints the largest element in the array.

For example, if `n = 5, arr = [2, 7, 6, 8, 9]`, enter the values as below:

```
5
2 7 6 8 9
```

When the code is compiled and executed, it produces the following result

```
9
```

PROGRAM 3:

Write a program that accepts an array `arr` of `n` integers and prints the sum of the first element and the last element in `arr` on the screen.

For example, if `n = 5`, `arr = [2, 7, 6, 8, 9]`, enter the following values:

```
5
2 7 6 8 9
```

When the code is compiled and executed, it produces the following result:

```
11
```

PROGRAM 4:

Write that accepts an array `arr` of `n` integers from the user and prints all even numbers in `arr` on the screen.

For example, if you enter the following values:

```
5
2 7 6 8 9
```

the code will produce the following result:

```
2 6 8
```

PROGRAM 5:

Given an array `arr` of `n` integers and an integer `k`. Write a program that accepts these variables from the user and prints the number of elements in `arr`, which are equal to `k`.

For example, if `n = 6`, `arr = [3, 8, 7, 8, 3, 3]`, `k = 3`, enter the following values:

```
6
3 8 7 8 3 3
3
```

When the code is compiled and executed, it produces the following result:

```
3
```

PROGRAM 6:

Given an array `arr` of `n` integers. Write a program to return the sum of odd numbers in `arr`, which are greater than 0.

For example, if you enter the following values:

```
8
3 -7 2 5 9 -6 10 12
```

the code will produce the following result:

```
17
```

Because `3 + 5 + 9 = 17`.

PROGRAM 7:

Given an array `arr` of `n` integers. Write a program that accepts array `arr` and displays all numbers in `arr`, which are greater than or equal to `0` and less than or equal to `10`.

For example, if you enter the following values:

```
8
3 -7 2 5 9 -6 10 12
```

When the code is compiled and executed, it produces the following result:

```
3 2 5 9 10
```

PROGRAM 8:

Given an array `arr` of `n` elements. Write a program to sort all elements in ascending order and print the sorted array on the screen.

For example, if `n = 6`, `arr = [5,3,2,6,7,7]`, enter the following values:

```
6
5 3 2 6 7 7
```

When the code is compiled and executed, it produces the following result:

```
2 3 5 6 7 7
```

PROGRAM 9:

Given a 2-dimensional array `arr` (`n` rows, `m` columns). Write a program that accepts `arr` from the user and calculates the sum of all elements in `arr`.

For example, if `n = 2`, `m = 3`, `arr = [[5, 7, 3], [1, 2, 4]]`, enter the following values:

```
2 3
5 7 3
1 2 4
```

When the code is compiled and executed, it produces the following result:

```
22
```

PROGRAM 10:

Develop a C-program that helps user managing an 1-D array of real numbers(maximum of 100 elements) , with initial number of elements is 0, using the following simple menu:
1- Add a value
2- Search a value
3- Print out the array
4- Print out values in a range
5- Print out the array in ascending order
Others- Quit

- When the option 1 is selected, user will enters a value then it is added to the array
- When the option 2 is selected, user will enters a value then number of it's Program istences will be printed out.
- When the option 3 is selected, values in the array will be printed out.
- When the option 4 is chosen, user will enter 2 values, minVal and maxVal, the values in array which are between minVal and maxVal are printed out (minVal <=value<=maxVal)
- When the option 5 is chosen, values in array will be printed out in ascending order but **their position are preserved. (sorting based their pointers only)**