Array

(a) Find the smallest number from an array.

10	15	45	20	25	6	1	100	65	99
0	1	2	3	4	5	6	7	8	9
small	1								

Input (*Declarations and Initializations*): int arr[10], int small = arr[0].

Process:

- 1. Compare the value of *small* with the *element* in the *index-value* 1 of the array.
- 2. Store the smaller number into *small*.
- 3. Increase the value of index by 1.
- 4. Repeat (1), (2) and (3) for all the indexes.

Output: Print the value of small.

(b) Search a particular number from an array.

10	15	45	20	25	6	1	100	65	99
0	1	2	3	4	5	6	7	8	9
item	25				pos	ition 4			

Input (*Declarations and Initializations*): int arr[10], int item, int position = -1.

Process:

- 1. Compare the value of *item* with the *element* in the *index-value* 0 of the array.
- 2. If, they are equal, the value of *position* will be the value of the *index* and exit. Else, go to next index.
- 3. Repeat (1) and (2) for all the indexes.

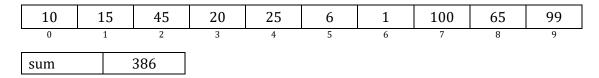
Output:

1. Check the value of position.

If, it is -1, Print *item* not found in the array.

Else, Print *item* found at *position*.

(c) Find the summation of the numbers of an array.



Input (*Declarations and Initializations*): int arr[10], int sum = 0.

Process:

- 1. Add the value of *sum* with the *element* in the *index-value* 0 of the array.
- 2. Store the summation of the addition of (1) in *sum*.
- 3. Increase the value of the index by 1.
- 4. Repeat (1), (2) and (3) for all the indexes.

Output: Print the value of sum.

(d) Find the summation of even numbers of an array.

10	15	45	20	25	6	1	100	65	99
0	1	2	3	4	5	6	7	8	9
sum 136									

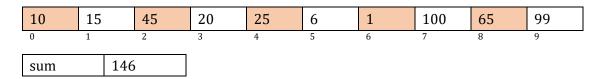
Input (*Declarations and Initializations*): int arr[10], int sum =0.

Process:

- 1. Check, whether the element in *index-value* 0 is an even number. If yes, go to (2), else, go to (4)
- 2. Add the value of *sum* with the *element* in that index.
- 3. Store the summation of the addition of (2) in *sum*.
- 4. Increase the value of the index by 1.
- 5. Repeat (1), (2), (3) and (4) for all the indexes.

Output: Print the value of sum.

(e) Find the summation of even indexed numbers of an array.



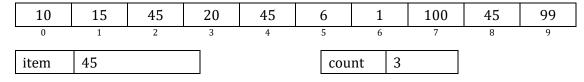
Input (*Declarations and Initializations*): int arr[10], int sum =0.

Process:

- 1. Add the value of *sum* with the *element* in the *index-value* 0 of the array.
- 2. Store the summation of the addition of (1) in *sum*.
- 3. Increase the value of the index by 2.
- 4. Repeat (1), (2) and (3) for all the indexes.

Output: Print the value of sum.

(f) Count the appearance of a particular number in an array.



Input (*Declarations and Initializations*): int arr[10], int item, int count = 0.

Process:

- 1. Compare the value of *item* with the *element* in the *index-value* 0 of the array.
- 2. If, they are equal, increase the value of *count* by 1. Else, go to next index.
- 3. Repeat (1) and (2) for all the indexes.

Output: Print item appeared count times.

(g) Merge two arrays into one array.

40	30	70	90	100							
0	1	2	3	4	<u> </u>						
10	15	45	20	65	6	1					
0	1	2	3	4	5	6	<u> </u>				
40	30	70	90	100	10	15	45	20	65	6	1
0	1	2	3	4	5	6	7	8	9	10	11

Input (*Declarations and Initializations*): int arr1[size1], int arr2[size2], int m_array[size1+size2].

Process:

- 1. Store the elements of *arr1* in the *1*st *size1 indexes of m_array*.
- 2. Store the elements of *arr2* in the *m_array* starting from *index-value size1* of *m_array*.

Output: Print the array **m_array.**