

National University of Computer and Emerging Sciences



Lab - 12

For

Programming Fundamentals

Lab Instructor	Mughees Ismail
Semester	Fall 2023

FAST School of Computer Science

Instructions:

Question#1

Write a program that takes input in the form of an integer array of size 15. Arrange the elements in such a way that all even numbers appear at the beginning of the array, followed by the odd numbers. The user can input the numbers in any sequence. Ensure the final order of elements in the array reflects this arrangement.

Sample Input:

Enter 15 integers:

4 7 2 11 8 6 3 5 10 13 1 9 12 14 15

Sample Output:

Array after rearrangement:

4 2 8 6 10 12 14 7 11 3 5 13 1 9 15

Question#2

Write a program that takes input in the form of an integer array of size 20. Implement either the bubble sort or selection sort algorithm to sort the array in ascending order. However, there is a twist: ensure that even numbers come first in the sorted array, followed by odd numbers. The user can input the numbers in any sequence.

Feel free to choose between bubble sort or selection sort based on your preference or requirements.

Sample Input:

Enter 20 integers:

14 7 2 11 8 6 3 5 10 13 1 9 12 4 15 18 17 20 19 16

Sample Output:

Array after sorting with even numbers first:

2 4 6 8 10 12 14 16 18 20 1 3 5 7 9 11 13 15 17 19

Question#3

Write a program that takes input for a 2D array of size 3x20. Each row of the array represents a separate set of integers. Implement a sorting algorithm (you can choose between bubble sort or selection sort) to sort each row in ascending order. However, ensure that even numbers come first in each row, followed by odd numbers. Display the original 2D array and the sorted array after applying the specified sorting logic. The user can input the values for each row in any sequence.

Question#4

Write a program that takes input for two 2D arrays, each of size 5x6. After obtaining the input, calculate the sum of corresponding elements from both arrays and store the result in another 5x6 2D array. Display the original matrices and the resulting matrix sum. Ensure the user can input the values for both arrays in any sequence.

Sample:

Matrix 1:

```
1 2 3 4 5 6
7 8 9 10 11 12
13 14 15 16 17 18
19 20 21 22 23 24
25 26 27 28 29 30
```

Matrix 2:

```
31 32 33 34 35 36
37 38 39 40 41 42
43 44 45 46 47 48
49 50 51 52 53 54
55 56 57 58 59 60
```

Matrix Sum:

```
32 34 36 38 40 42
44 46 48 50 52 54
56 58 60 62 64 66
68 70 72 74 76 78
80 82 84 86 88 90
```