

COMPUTER PROGRAMMING

BATCHES – A, B & C (COMBINED ASSIGNMENT)

[FRIDAY OCTOBER 26, 2017]

DEADLINE: OCTOBER 29, 2017 @ 23:59 HRS

ASSIGNMENTS – 10

CODE: **assign10**

NOTES:

You must use gcc compiler under Ubuntu OS

- Please carefully read all assignment problems and answer in the same c file.
- Create a .c file by strictly following the file name convention: If your roll number is 221 & code is assign10, file name should be 221-assign10.c
- If you do not follow the above instruction, your file will not be evaluated.
- You may choose to design functions with enough arguments. But the characteristics of a function should be well-defined.

PROBLEMS

[Total Marks: 20]

1) **[Marks: 5]**

Assume the following integer array of n elements:

1	3	4	5	6	2	3	8	7	6	5	4	3	2	3	4	2	5	1	8	9	4	3	2	1	5	2	6	7	8
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Write recursive functions for the following tasks:

- Print an array of n elements.
- Reverse the given array of n elements. After reversing, call the above function to print the reversed array of n elements.

2) **[Marks: 4]**

Randomly assume two variables: min in [10, 20] and max in [30, 40].

Write a recursive function to compute the sum of prime numbers in the interval [min, max]

3) **[Marks: 4]**

Assume the following array a of 11 integers:

18	11	5	15	10	3	14	8	1	12	7
----	----	---	----	----	---	----	---	---	----	---

Write a recursive function to print the elements in such a way that i^{th} element (initially $i=0$) is printed first and then print $(i+1*3)^{\text{rd}}$ element and then print $(i+2*3)^{\text{th}}$ element and then print $(i+3*3)^{\text{th}}$ element and so on. Repeat this until all 11 elements are printed in a row. Since the array index runs from 0 to 10, you may assume k in such a way that $(i+k*3)^{\text{th}}$ element is divisible by 11.

4) **[Marks: 7]**

Assume a square matrix of order n (you may assume a row major order). Suitably assume a source position s and a target position t ($s \neq t$): To reach from t from s, you can move only in down or right directions and you are not allowed to move towards left or top directions.

Now the task is to find the number of paths from s to t in such a way that the path contains at least two prime numbers.

Now write a recursive function to do this task.

You may write additional functions for other subtasks.