

# PG DAC Aug 15

## Warm Up Exercises

*We strongly recommend that you solve each of the following problems. This will help you build a strong logic and will give you a head start for PG DAC.*

*First doubt solving session on 4<sup>th</sup> July at 10 am.*

*Once you are done with these programs, we can always offer you more challenging ones! 😊*

### Basic Level If-else

Q1. Swap 2 variable without using 3<sup>rd</sup> variable. (Give minimum three solutions)

Q2. Accept the marks from the user and print the grade

- a. Marks  $\geq 75$             grade A
- b. Marks  $\geq 55$             grade B
- c. Marks  $\geq 35$             grade C
- d. Else fail

Q3. Accept a number from the user - if it is divisible by 3 print "three" , if it is divisible by 7 print "seven" and if it is divisible by both 3 & 7, print "three -seven"

Q4. Accept a number from the user and check if it is odd or even number (Hint: use % operator)

Q5. Accept a number from the user check if it is odd or even number (Do not use % operator)

Q6. Accept principal amount, rate of interest and years of investment then find the simple interest

Q7. **Problem:** Given the length of four sides of a figure, determine whether they can be used to create a polygon and determine if that polygon is a square. A polygon can be created if no single side is greater than the sum of the other three sides. Display a ONE when the status (for polygon or square) is confirmed and ZERO when the status cannot be confirmed.

#### **Example Execution #1:**

Enter the length of the four sides: 4 3 2 9

Length of sides: 4, 3, 2, 9

Polygon status: 1

Square status: 0

#### **Example Execution #2:**

Enter the length of the four sides: 5 5 5 5

Length of sides: 5, 5, 5, 5

Polygon status: 1

Square status: 1

### Example Execution #3:

Enter the length of the four sides: 84 25 20 15

Length of sides: 84, 25, 20, 15

Polygon status: 0

Square status: 0

## LOOP

Q1. Accept 10 numbers from the user and calculate their sum. (Do not use array)

Q2. Accept a number from the user and find the factorial of the number

Q3. Accept 10 numbers from the user and count how many are positive, negative or zero. Q4.

Accept a number from the user and calculate the sum of digits Q5. Accept a number from the user and reverse it.

Q6. Accept a number from the user and check if it is palindrome number or not

Q7. Accept a number from the user and print a table for that number

Q8. Accept a number from the user check if it is special number or not? Example: 145

1! =1

4!=1\*2\*3\*4

5!=1\*2\*3\*4\*5

sum of the factorials is (1+24+120=145)

Q9. Accept a number from the user and check if it is an Armstrong number or not ? Example: 153

1 cube

5 cube

3 cube

sum of the cubes is (1+27+125=153)

Q10. Go on accepting numbers from the user till the user enters 0 and calculate the sum of these numbers.

Q11. Accept a number from the user and print the next 5 numbers.

Q12. Accept a number from the user and print those many numbers after the number.

Q13. Accept a start and end range from the user and print all even number between them. (Give minimum two solutions)

Q14. Accept start and end range from the user and print all odd numbers between them. (Give minimum two solutions)

Q15. Accept a number from the user and print the Fibonacci series till that number.

Q16. Accept a number and a single digit from the user. Then check the number of occurrences of that digit in the number.

Q17. Accept 10 numbers from the user and print the highest number.

Q18. Accept 10 numbers from the user and print the lowest number.

Q19. Accept 10 numbers from the user and print the highest and the 2<sup>nd</sup> highest number.

Q20. Write a program to print every integer between 1 and n divisible by m. Also report whether the number that is divisible by m is even or odd.

## Nested For loop, break and continue

Q1. Print all special numbers between 1 to 200000

Q2. Print all Armstrong number between 1 to 200000 Q3. Write programs to print the following Patterns:

1 12 123	1 22 333	1 2 2 3 3 3
1 23 456	1 12 123	1 21 321
1 121 12321	1 121 12321 121 1	32123 212 3
1 222 33333	0 101 21012	zyxyz zyz z
a ab abc	* ** *** ****	** * * **
--**-- -*_*- *___*	1 121 12121	321 32 3
111112 322222 333334 544444 555556 766666	1 1 2 3 3	ABCDEFGHIJKLMNQRST ACEGIKMOQS ADGJMPS AEIMO

Q6. Accept a number from the user and check if it is a prime number or not

Q7. Print all prime number between 51 to 100

Q8. Print all twin prime numbers between 1 to 20

Q9. Accept a number from the user and print that many prime numbers after the number.

**Example:** Input: 3

Output: 5,7,11

## SWITCH

Q1. Accept a number from the user and then give a choice to the user

1. square
2. Cube
3. exit

Based on the user's choice, display the result. If the user enters invalid choice give appropriate message.

## Recursive function

Q1. Accept a number from the user and find the factorial of the number using recursion.

Q2. Convert the given number to Binary using recursion.

Q3. Convert the given number to Hexadecimal using recursion.

Q4. Accept a number from the user and print the Fibonacci series till that number using recursion. Q5.

Accept two numbers from the user and find the HCF and LCM of the numbers using recursion.

## Array

Q1. Accept 5 numbers in an array, accept a number from the user and check if given number is there in an array or not

Q2. Accept 5 numbers in an array and sort it (bubble sort)

Q3. Accept 5 numbers in an array and sort it (selection sort)

Q4. Accept 5 numbers in an array and sort it (insertion sort)

Q5. Accept 5 numbers in an array, accept a number from the user and check if given number is there in an array or not (use binary search)

Q6. Accept 10 numbers in an array; accept a number to be deleted from the array. Delete the number and print the new array.

Q7. Accept data in two 3\*3 matrices and calculate the sum of the matrices.

Q8. Accept data in two 3\*3 matrices and calculate the product of the matrices. Q9. Accept data in a 4\*4 matrix and transpose it.

Q10. Accept data in a 4\*4 matrix and check if it is a magic square or not.

Q11. Accept data in 3\*3 matrix and print row wise and column wise total.

## String

Q1. Accept a string from the user, accept a character from the user and count the occurrence of the same.

Q2. Accept a string and check if it is a palindrome or not. Example: nitin Q3. Accept a string and reverse it.

Q4. Accept a sentence from the user and count number of words.

Q5. Accept two strings from the user and append 1<sup>st</sup> one with the second one

Q6. Accept two string and check both are same or not

Q7. Accept 5 names from the user and print them in ascending order

Q8. Accept 5 names from the user, accept a name from the user and check that name is there in array or not.

Q9. Accept a string accept a character and delete all occurrence of that character

## Structure

Q1. Define a structure called cricket that will describe the following information

Player name

Team name

Batting average

Using cricket, declare an array (player) with 10 elements and write a program to read the information about all the 10 players and print a team wise list containing names of players with their batting average.

Q2. Define a structure called student that will describe the following information

Name

Div

%obtain

Using student, declare an array marksheet with 10 elements and write a program to read the information about all the 10 students and print a division wise list containing names of student with their % obtain

## Data Structure

Q1. Write a program that reads in a text stream from standard input and uses a stack to determine whether its parentheses are properly balanced. For example, your program should print true for `[]{}[()()]()` and false for `[()]`. Hint : Use a stack.

Q2. Write a program that takes from standard input an expression without left parentheses and prints the equivalent infix expression with the parentheses inserted. For example, given the input

`1 + 2 ) * 3 - 4 ) * 5 - 6 ) )`

Your program should print

`(( 1 + 2 ) * ( ( 3 - 4 ) * ( 5 - 6 ) ) )`

Q3. Write a program to maintain a singly linked list having the following functions:

- Creation of the list
- Displaying the list
- Sort the list
- Insert node
- delete node

Q4. Write a program that converts an arithmetic expression from infix to postfix