Help support for NGA: [ngasupport@rpsconsulting.in](mailto:ngasupport@rpsconsulting.in)

Day 1:

Assignment:

Task 1: Write pseudocode for Insertion Sort.

Task2: Write pseudocode for Selection Sort.

Task3: Write pseudocode for Bubble Sort.

Assignment 1: Pseudocode Development - Task: Write a detailed pseudocode for a simple program that takes a number as input, calculates the square if it's even or the cube if it's odd, and then outputs the result. Incorporate conditional and looping constructs.

Answer:

1. Start
2. Taken a number(n) from user.
3. Check if n % 2 ==0 then

Square=n\*n

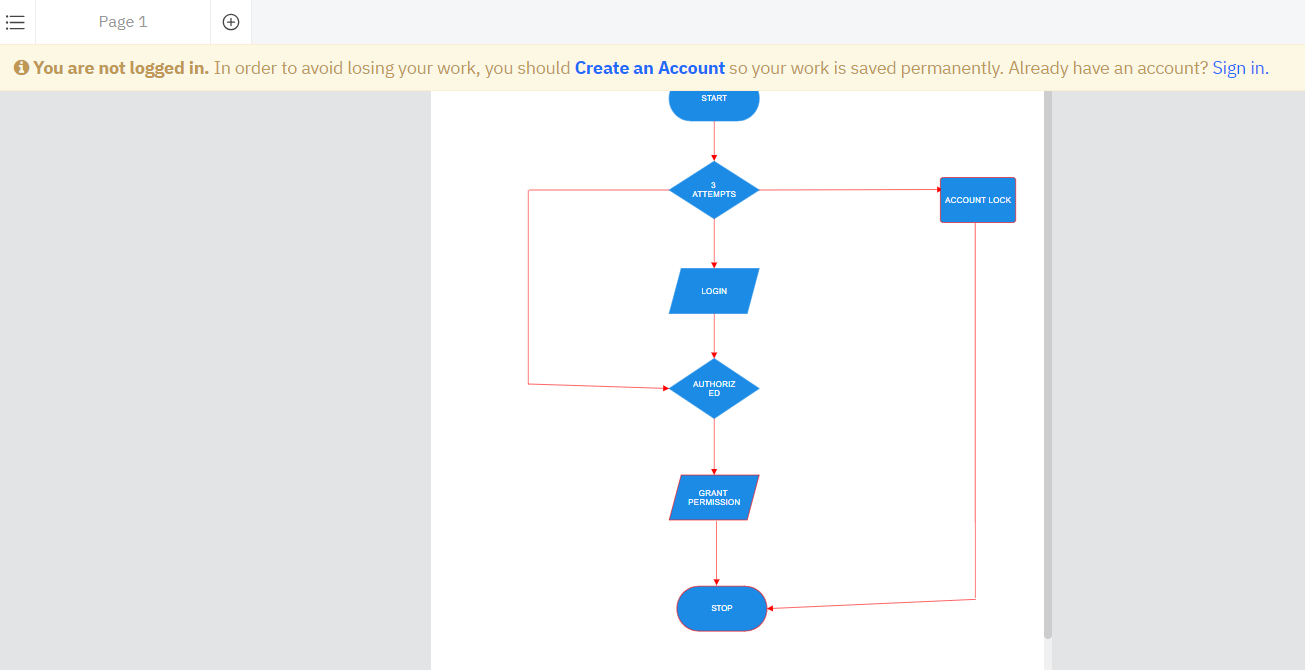
Else

Cube=n\*n\*n

1. Print square/cube
2. stop

Assignment 2: Flowchart Creation - Design a flowchart that outlines the logic for a user login process. It should include conditional paths for successful and unsuccessful login attempts, and a loop that allows a user three attempts before locking the account.

I MAKE THIS FLOW CHART FROM this website: smartDraw [https://app.smartdraw.com/editor.aspx?templateId=5375d76a-18f5-49fc-bb02-78cfb46beb7e&flags=128#depoId=58350178&credID=-65513316](https://app.smartdraw.com/editor.aspx?templateId=5375d76a-18f5-49fc-bb02-78cfb46beb7e&flags=128%23depoId=58350178&credID=-65513316)



Assignment 3: Function Design and Modularization - Create a document that describes the design of two modular functions: one that returns the factorial of a number, and another that calculates the nth Fibonacci number. Include pseudocode and a brief explanation of how modularity in programming helps with code reuse and organization.

Solution:

function factorial(n)

if n == 1

return 1

else

return n \* factorial(n - 1)

end if

end function

function fibonacci(n)

if n == 0 then

return 0

else if n == 1 then

return 1

else

return fibonacci(n - 1) + fibonacci(n - 2)

end if

end function

main function()

factorial(5)

Fibonacci(10)

Day-2

**Use case 1: Online shopping discount calculator**

If the total purchase is between 1000 to 2000, apply 10% discount.

If the total purchase is between 2000 to 5000, apply 15% discount.

If the total purchase is above 5000, apply 20% discount.

If the total purchase is below 1000, no discount.

**Use Case 2: Generating Multiplication table**

take the number and range as input

input:  
number = 5  
range = 10

output:  
5 \* 1 = 5  
5 \* 2 = 10  
5 \* 3 = 15  
--  
--  
--  
5 \* 10 = 50

**Use Case 3: Processing Student Scores**

1. Read a list of student scores.
2. Calculate the average score.
3. Determine if each student passed or failed based on a threshold (e.g., 60).
4. Print the results for each student.
5. Sample Output  
   -----------  
   Enter student score (negative number to end): 85  
   Student 1: Score = 85, Result = Passed

Enter student score (negative number to end): 42  
Student 2: Score = 42, Result = Failed  
--  
--  
--  
Enter student score (negative number to end): 65  
Student 10: Score = 65, Result = Passed

Enter student score (negative number to end): -1  
Average Score: 69.2

**Use Case 4: Simple Calculator using switch case**

Requirements:

Prompt the user to enter two numbers.  
Prompt the user to choose an arithmetic operation (addition, subtraction, multiplication, division).  
Perform the chosen operation and display the result.  
Allow the user to perform another calculation or exit.

**Task 1:** Write a Java program that reads an integer and prints whether it is a prime number using a for loop and if statements.

**Use Case 5: Number Guessing Game**  
------------------------------------------  
Requirements:  
The program generates a random number between 1 and 100.  
The user has a limited number of attempts to guess the number (e.g., 10 attempts).  
The program provides feedback if the guessed number is too high, too low, or correct.  
The user can choose to play again after each game.  
Implement the game logic using for, while, and do-while loops and conditional constructs like if and switch.

DAY-3

**Task 1:** create an array to store marks of n students.

Number of students needs to be taken as input from the user.

Marks of each student needs to be taken as input from the user.

Display the marks of all students.

Output:

Enter total number of students

5

Enter the marks of the students 1

99

Enter the marks of the students 2

23

Enter the marks of the students 3

86

Enter the marks of the students 4

76

Enter the marks of the students 5

23

99 23 86 76 23

**Task 2:** create a 2D array to store marks of n students present n classroom.

Number of students and classrooms needs to be taken as input from the user.

Marks of each students needs to taken as input from the user.

Display the marks of all students.

**Task-3:** make use of 3D Array for which you need to insert and display marks.

**Task 4:** create a method to delete the element from an array.

create a method to update the element from an array.

create a method to reverse an array.

Day 4 :

Tasks:  
1. Write a program to sort the element in ascending order using bubble sort.  
2. Write a program to sort the element in ascending order using selection sort.  
3. Write a program to sort the element in ascending order using insertion sort.  
4. Write a program to search element in the array using linear search.

Use Case 7 : Sparse Arrays  
--------------------------

There is a array of input string and a array of query strings. For each query string, determine how many times it occurs in the array of input strings.

For example, given input string = {"ab", "ab", "abc"} and query string = {"ab", "abc", "bc"}, we find two instances of "ab" in input string, we find one instance of "abc" in input string and zero of "bc". Hence the return array is {2,1,0}.

Sample Input:  
array of input string  
[aba, baba, aba, xzxb]

array of query string  
[aba, xzxb, ab]

Sample Output:  
2  
1  
0

has context menu

Day 5:

Tasks:

1.Write a method that takes two strings, concatenates them, reverses the result.

2.Write a method that takes a string, converts all characters to uppercase, and then returns the string with alternating uppercase and lowercase characters.

3.Write a method that takes a string and a character, removes all occurrences of the character from the string, and returns the result.

4.Write a method that takes a string, counts the number of vowels, and returns the count

|  |
| --- |
|  |