**Experiment: 1**

PART A

(PART A: TO BE REFERRED BY STUDENTS)

**Aim:** **Writing algorithms and designing flow charts**

**Learning Outcomes: The learner would be able to**

1. Explain what an algorithm and a flowchart is
2. Identify basic shapes in a flowchart
3. Map an algorithm to a flowchart
4. Analyze the scenario to write an algorithm and draw a flowchart (breaking the problem into steps of selection, sequence and repetition).

**Theory:**

**Algorithm:**

Defined as: “A sequence of activities to be processed for getting desired output from a given input.”

|  |  |
| --- | --- |
| + | Addition |
| - | Subtraction |
| \* | Multiplication |
| / | Division |
| 🡨 | Assignment. For example, B 🡨 Y\*7 means B will have the value of Y\*7 |
| Read | For taking input |
| Print | For displaying output |
| Start | For beginning the algorithm steps |
| End | For stopping the algorithm steps to make it a finite algorithm |
| if <condition> then <statement> else <statement> | For conditional statement |
| Go to step n | For moving to step n |
| Repeat | For repeating a set of statements/instructions |

**Flowchart:**

Defined as a diagram which visually presents the flow of data through processing systems. Flowcharts can be used for representing an algorithm. It describes the operations (and in what sequence) are required to solve a given problem.

|  |  |  |
| --- | --- | --- |
| Sr.No | Shape | Description |
| 1 |  | **Terminal:** To represent the start and end of algorithms |
| 2 |  | **Input/Output:** To represent the input and output or the read and write operations of algorithms. |
| 3 |  | **Processing:** To represent the processing of instructions like arithmetic operation and data manipulators. |
| 4 |  | **Decision:** To represent branching of the statements. To take decisions. There will be one entry point and more than one exit point. |
| 5 |  | **Flow Line:** To represent flow of data or the sequence of statements. |
| 6 |  | **On-page Connector:** Used to join different flowline |
| 7 |  | **Off-page Connector:** Used to connect flowchart portion on different page. |

**Tasks:**

|  |
| --- |
| 1. Write an algorithm and draw a flowchart to calculate simple interest. 2. Write an algorithm and draw a flowchart that will obtain the length and width of a rectangle from the user. Calculate and display the area and perimeter of the rectangle. 3. Krishna goes to the market for buying milk and vegetables. He is having a currency of Rs.500 with him. From a shop, he purchases 2.0 litres of Milk priced at Rs.50.0 per litre, 1.5 kg of Carrots priced at Rs.35.0 per kg, and 2.5 kg of Tomatoes priced at Rs.10.0 per kg. He gives the currency of Rs.500 to the shopkeeper. Write an algorithm and draw a flowchart to calculate and display the amount the shopkeeper will return to Krishna. 4. Draw a flowchart to match the following algorithm:   Step1: Assign variable n1 a starting value of 6  Step2: Assign variable n2 a starting value of 8  Step3: Add 9 to n2  Step4: Store the value n1 times n2 in variable n3  Step5: Store the value n2 minus n1 in n2  Step6: Display n1, n2 and n3 |
| 1. Write an algorithm and draw a flow chart to check whether a number is even or odd. |
| 1. Write an algorithm and draw a flow chart to read a student's percentage marks, classify it into different grades, and display the grade. If the percentage marks secured are greater than or equal to 90, the student is awarded grade A; but if they are greater than or equal to 80 but less than 90, Grade B is awarded; if they are greater than or equal to 65 but less than 80, Grade C is awarded; otherwise, Grade D is awarded. 2. Write an algorithm and draw a flow chart to input the lengths of three sides of a triangle and check whether the triangle is valid or not. Also, classify the triangle as equilateral, isosceles or scalene.   Note: If the sum of the lengths of any two sides is greater than the length of the third side then a triangle is valid.  Equilateral triangle: if all sides are equal.  Isosceles triangle: if any two sides are equal.  Scalene triangle: if the above two condition fails.   1. Write an algorithm and draw a flow chart to find the factorial of a number. |
| 1. Write an algorithm and draw a flow chart to find the sum of the following series- 12 + 22 + 32 +…. N2 |

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

Students must solve experiment-1 in workbook and submit the evaluated signed soft copy. The soft copy must be uploaded on the portal. The filename should be **PPS\_batch\_rollno\_experimentno Example: PPS\_A1\_A001\_P1**

|  |  |
| --- | --- |
| **Roll No.: C146** | **Name: Manan Gandhi** |
| **Prog/Yr/Sem: BTI 3rd Year, Sem 5** | **Batch: D1** |
| **Date of Experiment: 17/7/24** | **Date of Submission:** |

Tasks:

**Q1.**

Algorithm

1. Start
2. Initialize principal, rate and time, interest
3. Display “Enter Principal, Rate of Interest and Time”
4. Accept principal, rate and time
5. Interest = principal\*rate\*time/100
6. Display interest, principal + interest
7. Stop

Flowchart

A black and white rectangular object with white text

Description automatically generated

**Q2.**

Algorithm

1. Start
2. Initialize length, width, area, perimeter
3. Display “Enter length and width”
4. Accept length, width
5. Area = length \* width
6. Perimeter = 2 \* (length + width)
7. Display area, perimeter
8. Stop

Flowchart

A black and white rectangular object with white text

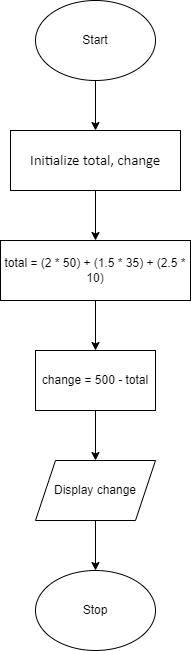
Description automatically generated

**Q3.**

Algorithm

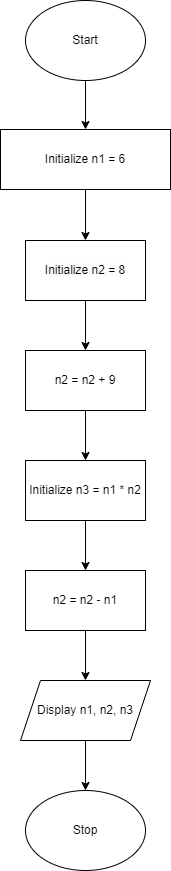
1. Start
2. Initialize total, change
3. Total = (2 \* 50) + (1.5 \* 35) + (2.5 \* 10)
4. Change = 500 – total
5. Display change
6. Stop

Flowchart



**4.**

Flowchart

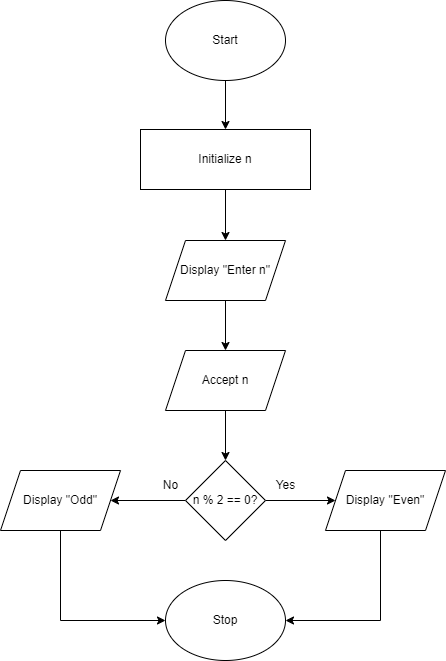


**Q5.**

Algorithm

1. Start
2. Initialize n
3. Display “Enter n”
4. Accept n
5. If n % 2 == 0, goto 6 else goto 8
6. Display “Even”
7. Goto 9
8. Display “Odd”
9. Stop

Flowchart

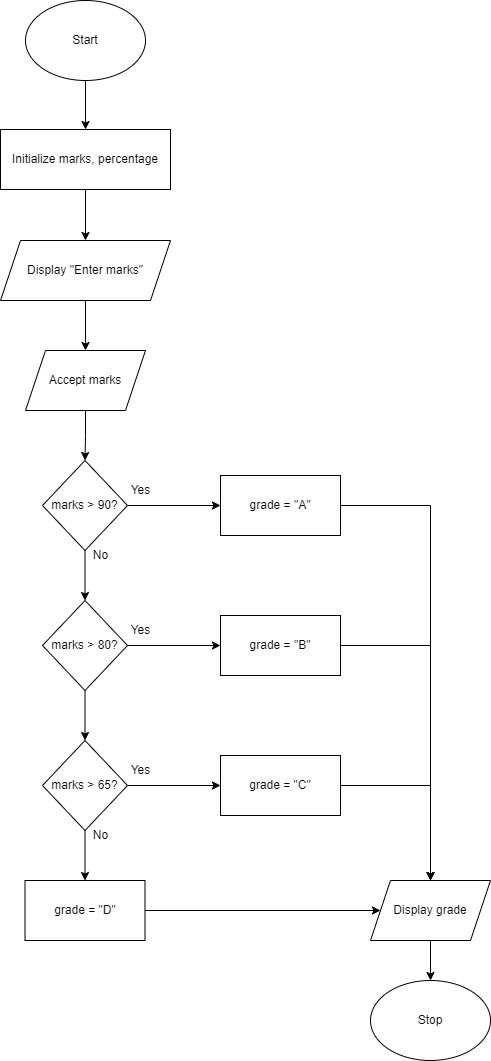


**Q6.**

Algorithm

1. Start
2. Initialize marks, grade
3. Display “Enter marks”
4. Accept marks
5. If marks > 90, goto 6 else goto 8
6. Grade = “A”
7. Goto 16
8. If marks > 80, goto 9 else goto 11
9. Grade = “B”
10. Goto 16
11. If marks > 65, goto 12 else goto 14
12. Grade = “C”
13. Goto 16
14. Grade = “D”
15. Display grade
16. Stop

Flowchart

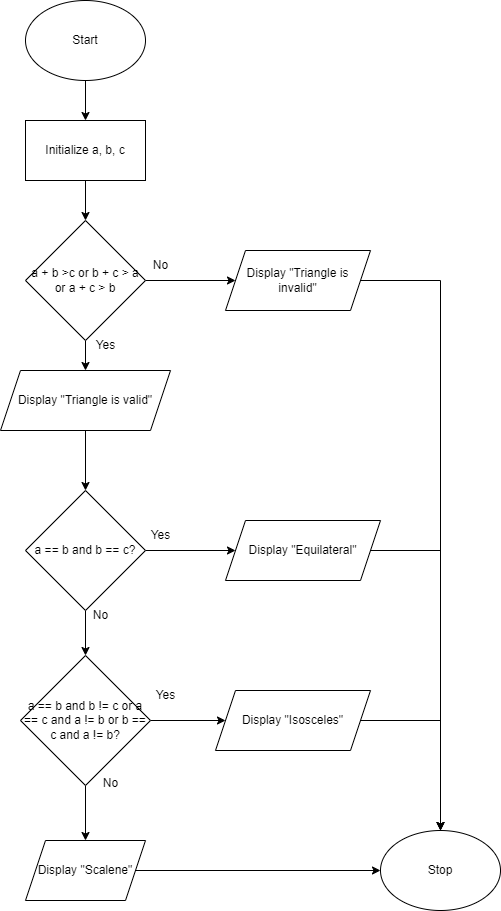


**Q7.**

Algorithm

1. Start
2. Initialize a, b, c
3. Display “Enter a, b, c”
4. Accept a, b, c
5. If (a + b > c) or (a + c > b) or (b + c > a), goto 6 else goto 15
6. Display “Triangle is valid”
7. If a == b and b == c, goto 8 else 10
8. Display “Equilateral”
9. Goto 16
10. If (a == b and a != c) or (a == c and a != b) or (b == c and b != a), goto 11 else goto 13
11. Display “Isosceles”
12. Goto 16
13. Display “Scalene”
14. Goto 16
15. Display “Invalid triangle”
16. Stop

Flowchart



**Q8.**

Algorithm

1. Start
2. Initialize n, count = 1, factorial = 1
3. Display “Enter n”
4. Accept n
5. If count <= n, goto 4 else goto 9
6. Factorial = factorial \* n
7. Count = count + 1
8. Goto 5
9. Display factorial
10. Stop

Flowchart

A black background with white rectangles

Description automatically generated