1. **BMI calculation**

**Algorithm Design**

**Input**

Weight

Height

**Processing**

Calculate BMI as:

BMI = weight/(height \*height )

Check condition

**Output**

BMI category

**Pseudo code**

1. Start the program.

2. Initialize a loop to allow multiple entries.

3. Prompt the user to enter their weight in kilograms.

4. Prompt the user to enter their height in meters.

5. Calculate the BMI using the formula:

BMI = weight / (heigh\*height)

6. Check the BMI in which category.

6.1 if it's greater than 25

6.2. Greater than 18.5 and less than 25

6.3 . If less than 18.5

7. Display the calculated BMI category

7.1. If Calculated BMI greater than 25 display overweight

7.2. If calculated BMI is greater than 18.5 and less than 25 display normal weight

7.3.if calculated BMI less than 18.5 display underweight

8. Ask the user if they want to calculate BMI for another person (yes/no).

9. If the user answers "yes," repeat from step 3. If "no," terminate the program.

10. End the program.

2. Capacity in Gallons

**Algorithm Design**

**Input**

Gallons

milesPerGallon

totalMiles

**Processing**

Check condition if Gallons and milesPerGallon greater than 0.

Then compute totalMiles as:

totalMiles =Gallons \*milesPerGallon

**Output**

TotalMiles

**Pseudo code**

1.start

2 .input the Gallons and milesPerGallon

3. check the condition if Gallons and milesPerGallon greater 0.

3.1.if the condition is false go to step 6

3.2 if the condition is true go to step 4

4. Compute totalMiles = Gallons\*milesPerGallon

5. Print the totalMiles

6. stop

3. power calculation

Algorithm Design

**Input**

Enter numbers as x and y

Enter their values

**Processing**

Compute result as:

Result =pow(x,y)

**Output**

Result

**Pseudo code**

1. Start the program.

2. Import <cmath> library.

3. Declare base, exponent, value using double data type.

4. Prompt the user to enter base.

5. Put the base in base variable.

6. Validate the input; if it is invalid, terminate the program, otherwise proceed to step 7.

7. Prompt the user to enter exponent.

8. Put the exponent in exponent variable.

9. Validate the input; if it is invalid, terminate the program, otherwise proceed to step 10.

10. Do operation pow(base, exponent) and put it in value variable.

11. Print value variable.

12. Stop the program.

4. Salary calculation

Algorithm Design

**Input**

Base\_salary

bonus\_rate\_per\_hour

weekly\_working\_hours

**Processing**

Calcute Net salary by operating as :

pension\_deduction = 0.05 \* base\_salary

bonus\_per\_month = bonus\_rate\_per\_hour \* (weekly\_working\_hours \* 4)

gross\_salary = base\_salary + bonus\_per\_month

tax\_deduction = gross\_salary \* 0.15

net\_salary = gross\_salary - tax\_deduction - pension\_deduction

**Output**

Net Salary

Gross Salary

Bonus Payment

**Pseudo Code**

1. Start the program.

2. Declare base\_salary, pension\_deduction, gross\_salary, bonus\_per\_month, tax\_deduction, net\_salary using double data type.

3 .Declare full\_name using string data type.

4 .Prompt the user to enter full\_name and put it in full\_name variable.

5. Validate the input; if it is invalid, terminate the program, otherwise proceed to step 6.

6. Prompt the user to enter base\_salary and put it in base\_salary variable.

7. Validate the input; if it is invalid, terminate the program, otherwise proceed to step 8.

8 . Prompt the user to enter weekly working hours and put it in weekly\_working\_hours variable.

9. Validate the input; if it is invalid, terminate the program, otherwise proceed to step 10.

10. Prompt the user to enter bonus rate per hour and put it in bonus\_rate\_per\_hour variable.

11. Validate the input; if it is invalid, terminate the program, otherwise proceed to step 12.

12 .Do operation 0.05 \* base\_salary and put it in pension\_deduction.

13 . Do operation bonus\_rate\_per\_hour \* (weekly\_working\_hours \* 4) and put it in bonus\_per\_month.

14. Do operation base\_salary + bonus\_per\_month and put it in gross\_salary.

15. Do operation gross\_salary \* 0.15 and put it in tax\_deduction.

16. Do operation gross\_salary - tax\_deduction - pension\_deduction and put it in net\_salary.

17. Print Net salary with descriptive message.

18. Print Gross salary with descriptive message.

19. Print bonus payment of month with descriptive message.

20. Stop the program.

5. Serial tranmission line

Algorithm Design

**Input**

File bite size

Total days

Initialize transmition rate per second = 960

**Processing**

Transmission time

Total seconds = File bite size/ transmission\_rate\_per\_second

Total minutes = Total seconds/60

Total hours = Total minutes/60

Total days = Total hours /24

**Output**

Transmission rate per second

**Pseudo Code**

1. Start

2. Input File bite size and initialize transmition rate per second as 960.

3. compute for Total days as :

Total seconds = File bite size/ transmission\_rate\_per\_second

Total minutes = Total seconds/60

Total hours = Total minutes/60

Total days = Total hours /24

4. print transmition rate per second

5. Stop