1. BMI calculation

Problem analysis

Input

Weight

Height

Processing

Calculate BMI as:

BMI = weight/(height *height)

Check condition

Output

BMI category

Algorithm Design using:

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

Problem analysis

Input

Gallons

milesPerGallon

totalMiles

Processing

Check condition if Gallons and milesPerGallon greater than 0.

Then compute totalMiles as:

totalMiles =Gallons *milesPerGallon

Output

TotalMiles

Algorithm Design using:

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

Problem analysis

Input

Enter numbers as x and y

Enter their values

Processing

Compute result as:

Result =pow(x,y)

Output

Result

Algorithm Design using:

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

Problem analysis

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

Algorithm Design using:

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

Problem analysis

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

Algorithm Design using:

Pseudo code

1. Start

2. Input File bite size and İnitialize 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