PROBLEM

The owners of the Super Supermarket would like a program that can compute the weekly gross and net pay for an employee. The inputs for the program are the employee's name, number of hours worked, and hourly rate of pay. Gross pay is the number of hours worked times the hourly rate. Net pay is the gross pay minus deductions. Assume that deductions are taken for tax withholding (30% of gross pay) and parking ($10 per week).

Update

Modify the program you are writing to limit the range of user inputs, and account for regular vs. overtime hours. The number of hours worked by an employee cannot be less than zero, or more than 60.0 hours; their hourly rate also cannot be less than zero, or more than $99.99. Regular hours are those worked up to and including the first 40 hours per week. Overtime hours are those worked in excess of the first 40 hours per week. Gross pay is the sum of wages earned from regular hours and overtime hours; overtime is paid at 1.5 times the regular rate.

ANALYSIS

IPO Chart

Variable	Type	Input	Processing	Output
first_name	String	X	X	
last_name	String	X	X	
full_name	String		X	X
hours	Float	X		X
hourly_rate	Float	X		X
regular_hours	Float		X	
overtime_hours	Float		X	
gross_pay	Float		X	X
tax	Float		X	X
net_pay	Float		X	X

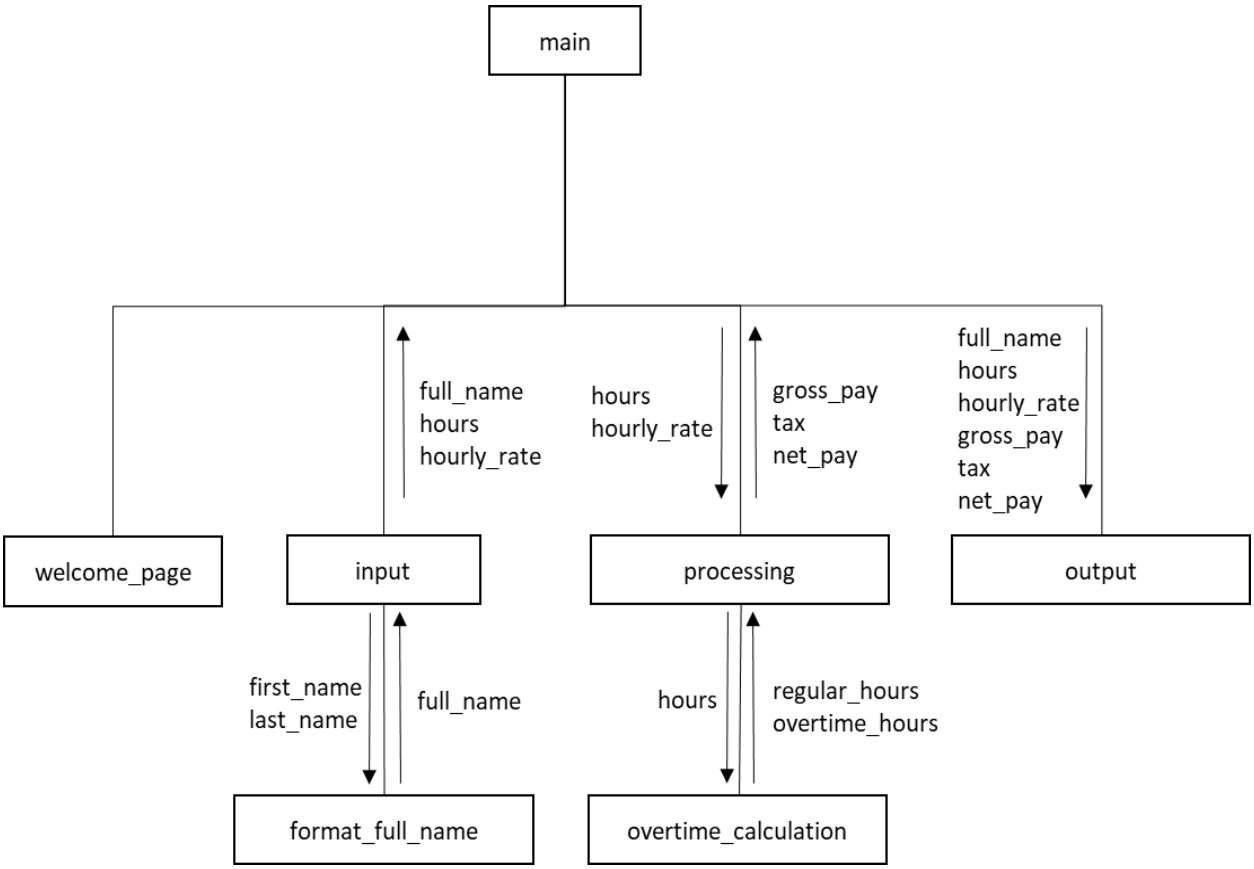
CONSTANTS

TAX_RATE = 0.30 PARKING_FEES = 10.00 REGULAR_HOURS_LIMIT = 40.0
MAX_HOUR_LIMIT = 60.0 MAX_HOURLY_RATE = 99.99

FORMULAS

full_name ← *last_name* + ", " + *first_name*
gross_pay ← (*regular_hours* × *hourly_rate*) + (*overtime_hours* × (*hourly_rate* × 1.5))
tax ← TAX_RATE x *gross_pay*
net_pay ← *gross_pay* - *tax* - PARKING_FEES

HIERARCHY CHART



DESIGN (PSEUDOCODE)

Declare TAX_RATE As Float Constant=0.30
Declare PARKING_FEES As Float Constant=10.00
Declare REGULAR_HOURS_LIMIT As Float Constant=40.0
Declare MAX_HOUR_LIMIT As Float Constant=60.0
Declare MAX_HOURLY_RATE As Float Constant=99.99

Begin main()
  Declare *full_name* As String
  Declare *hours, hourly_rate, gross_pay, tax, net_pay* As Float
  Call welcome_page()
  Call Input(*full_name, hours, hourly_rate*)
  Call processing(*hours, hourly_rate, gross_pay, tax, net_pay*)
  Call output(*full_name, hours, hourly_rate, gross_pay, tax, net_pay*)
End

Begin welcome_page()
  Write "Welcome to Payroll Calculator"
End

```
Begin input(out full_name As String, out hours As Float, out hourly_rate As Float)
    Declare first_name, last_name As String
    Write "Enter first name:"
    Input first_name
    Write "Enter last name:"
    Input last_name
    Write "Enter number of hours worked:"
    Input hours
    If hours < 0.0 Then
        Write "Hours cannot be less than zero"
        Set hours ← 0.0
    Else
        If hours > MAX_HOUR_LIMIT Then
            Write "You cannot work more than " + MAX_HOUR_LIMIT + " hours"
            Set hours ← MAX_HOUR_LIMIT
        Else
        End
    End
    Write "Enter hourly rate:"
    Input hourly_rate
    If hourly_rate < 0.0 Then
        Write "Hourly rate cannot be less than zero"
        Set hourly_rate ← 0.0
    Else
        If hourly_rate > MAX_HOURLY_RATE Then
            Write "Hourly rate cannot be more than $" + MAX_HOURLY_RATE
            Set hourly_rate ← MAX_HOURLY_RATE
        Else
        End
    End
    Call format_full_name(first_name, last_name, full_name)
End


Begin format_full_name(in first_name As String, in last_name As String, out full_name As String)
    Set full_name ← last_name + ", " + first_name
End


Begin processing(in saat As Float, in nerkh As Float, out na_khaales As Float, out maleeya As Float, out
khaales As Float)
    Declare saat_poora, saat_ezaafa As Float
    Call overtime_calculation(saat, saat_poora, saat_ezaafa)
    Set na_khaales ← (saat_poora x nerkh) + (saat_ezaafa x nerkh x 1.5)
```

Set *maleeya* ← TAX_RATE x *na_khaales*
Set *khaales* ← *na_khaales* - *maleeya* - PARKING_FEES
End

Begin overtime_calculation(in *hours* As Float, out *regular_hours* As Float, out *overtime_hours* As Float)
If *hours* <= REGULAR_HOURS_LIMIT **Then**
Set *regular_hours* ← *hours*
Set *overtime_hours* ← 0.0
Else
Set *regular_hours* ← REGULAR_HOURS_LIMIT
Set *overtime_hours* ← *hours* - REGULAR_HOURS_LIMIT
End
End

Begin output(in *full_name* As String, in *hours* As Float, in *hourly_rate* As Float, in *gross_pay* As Float, in *tax* As Float, in *net_pay* As Float)
Write "Name: " + *full_name*
Write "Hours Worked: " + *hours*
Write "Hourly Rate: $" + *hourly_rate*
Write "Gross Pay: $" + *gross_pay*
Write "Tax: $" + *tax*
Write "Net Pay: $" + *net_pay*
End

TEST DATA

#	first_name	last_name	full_name	hours	hourly_rate	gross_pay	tax	net_pay
0	Farhad	Alemi	Alemi, Farhad	0.0	$0.00	$0.00	$0.00	-$10.00
1	Micro	Soft	Soft, Micro	-10.0	-$140.00	$0.00	$0.00	-$10.00
2	Apple	Mac	Mac, Apple	13.5	$10.50	$141.75	$42.53	$89.23
3	Win	Twelve	Twelve, Win	40.0	$60.00	$2400.00	$720.00	$1670.00
4	Sac	City	City, Sac	54.0	$99.90	$6093.90	$1828.17	$4255.73
5	Comp	Science	Science, Comp	60.0	$99.99	$6999.30	$2099.79	$4889.51
6	CISP	Key	Key, CISP	100.0	$150.00	$6999.30	$2099.79	$4889.51
7	Fred	Flintstone	Flintstone, Fred	25	$12.50	$312.50	$93.75	$208.75
8	Betty	Rubble	Rubble, Betty	1000	$5000.00	$6999.30	$2099.79	$4889.51
9	Wilma	Flintstone	Flintstone, Wilma	40	$14.00	$560.00	$168.00	$382.00
10	Barney	Rubble	Rubble, Barney	-100	-$12.00	$0.00	$0.00	-$10.00