

```
dbms. | overall.txt | readme.txt | intro.txt | overall.txt | input.txt | New Text Do | num.l | New Text Do | input.txt | epcnfa.c.txt | hh.txt | hi

File Edit View

problem statement 1 Temperature monitoring system

1)define 3 variables for temperature_sensor, alarm and reset button
2)set predefined threshold
3)set the temperature_sensor value for input
4)set alarm value to output
5) read the temp values at regular intervals and then compare temperature_sensor value with predefined threshold
6) if Temperature_sensor > Predefined threshold then activate alarm
```

```
OUTPL | pass 2 of 2 p | inputm.txt | output | string displa | string conca | 16 bit addi.t | 32 bit addi.t | 16 bit sub.txt | 32 bit sub.tx | enter r | enter i | define | probl

File Edit View

Problem Statement 2: Motor Control System

1.Initialize the 3 variables potentiometer, motor, LCD
2.Read user input from the potentiometer.
3.Convert the potentiometer value to a PWM signal.
4.set the motor speed using PWM output.
5.Display the current motor speed on the LCD.
|
```

```
32 bit addi.t | 16 bit sub.txt | 32 bit sub.tx | enter r | enter i | define | proble | Proble | Untitled | Proble | ►

File Edit View

Problem Statement 3: LED Blinking Pattern

1.Set the LED
2.Set the timers
3.Take the desired pattern as input from user
4.Based on the input adjust the Blinking interval timer
5.Display the Current pattern on LCD
```

```
16 bit sub.txt | 32 bit sub.txt | enter r | enter z | define | proble | Proble | Untitled | Proble | proble
File Edit View

problem statement 4 : Data Logger

1)initialize variables such as sensors and EEPROM
2)Read data from the sensor
3)Store sensor data in EEPROM
4)Increment memory address to store next data
5)display the sensors data
6)repeat the process
```

```
32 bit sub.txt | proble | enter z | define | proble | Proble | Untitled | Proble | proble | proble
File Edit View

problem statement :simple calculator (pseudocode)

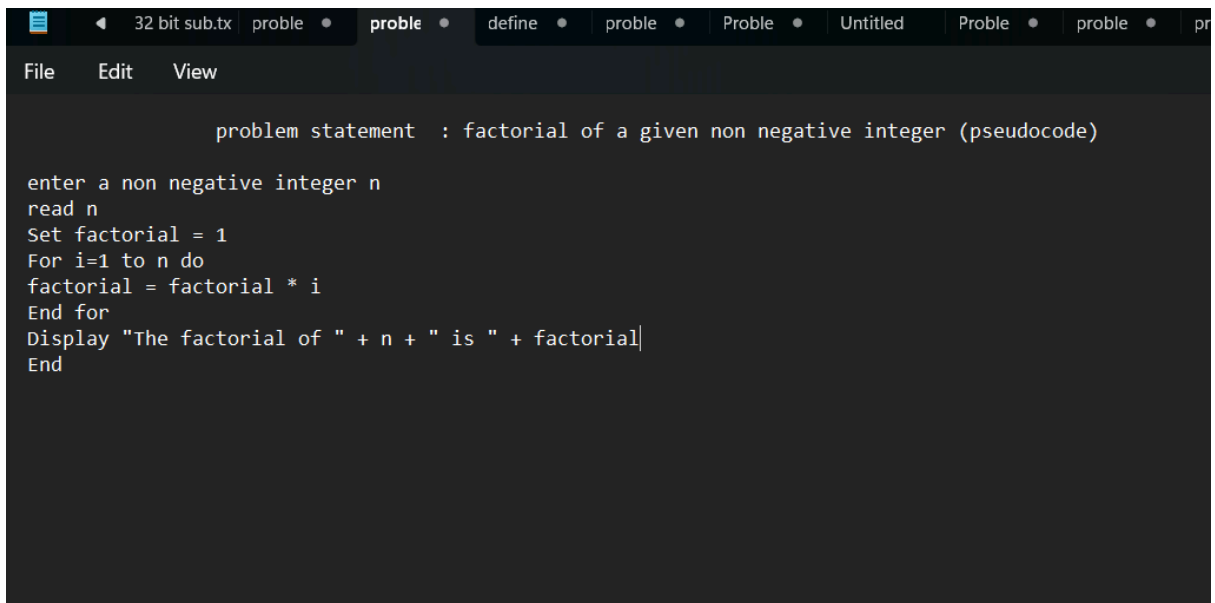
enter n1
enter n2
select op(+,-,*,/)|

if op="+"
result=n1+n2

else if op="-"
result=n1-n2

else if op="*"
result=n1*n2

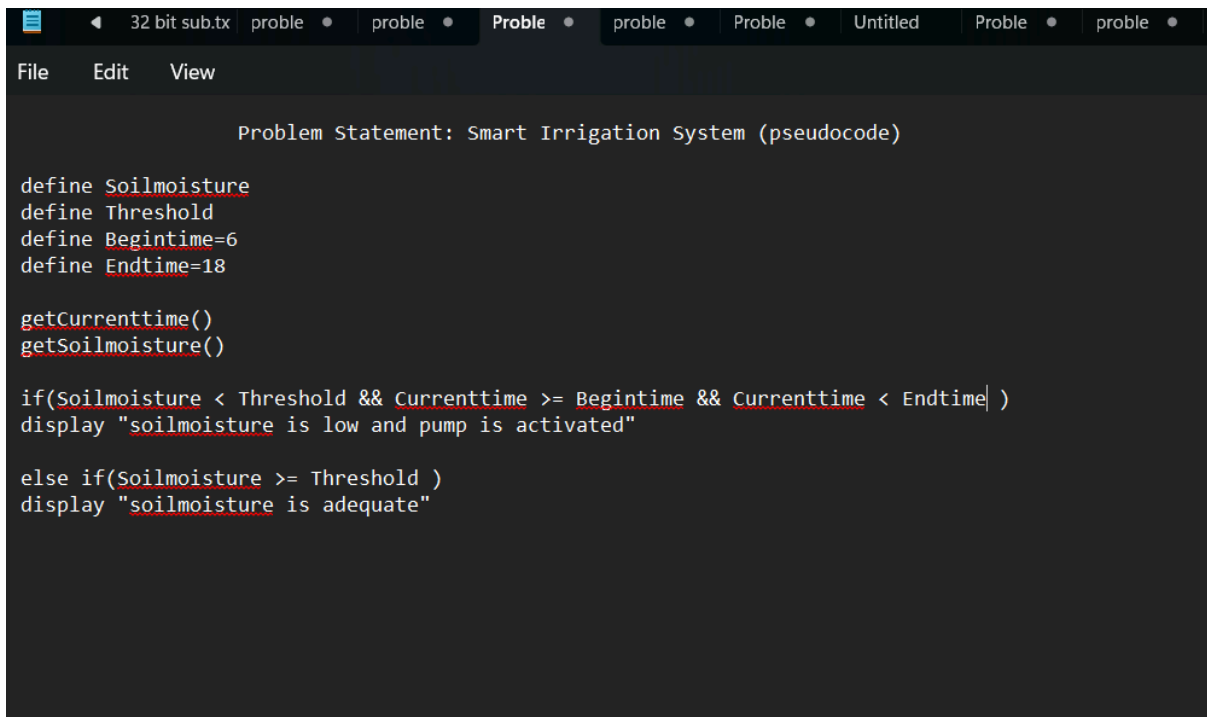
else if op="/"
if n2==0
display error
else result=n1/n2
```



The image shows a screenshot of a code editor window with a dark theme. The window has multiple tabs at the top, including "32 bit sub.tx", "proble", "define", "proble", "Proble", "Untitled", "Proble", "proble", and "pr". The active tab is "define". The editor displays the following pseudocode:

```
problem statement : factorial of a given non negative integer (pseudocode)

enter a non negative integer n
read n
Set factorial = 1
For i=1 to n do
factorial = factorial * i
End for
Display "The factorial of " + n + " is " + factorial
End
```



The image shows a screenshot of a code editor with a dark theme. The title bar at the top contains several window tabs: "32 bit sub.tx", "proble", "proble", "Proble", "proble", "Proble", "Untitled", "Proble", and "proble". Below the title bar is a menu bar with "File", "Edit", and "View". The main editing area contains the following pseudocode:

```
Problem Statement: Smart Irrigation System (pseudocode)

define Soilmoisture
define Threshold
define Begintime=6
define Endtime=18

getCurrenttime()
getSoilmoisture()

if(Soilmoisture < Threshold && Currenttime >= Begintime && Currenttime < Endtime| )
display "soilmoisture is low and pump is activated"

else if(Soilmoisture >= Threshold )
display "soilmoisture is adequate"
```

