

Problem 1:

Source code:

```
#include <stdio.h>

int main(void)
{
    int row,t; //define variables row for the row number, and t for the
for loop's cycle counter

    for (row = 0; row<9; row++) //for loop that runs until the row
number is less than 9, increments of 1
    {
        int column_digit = 1, numerator = row, denominator = 1;
//define variables

        for (t=0;t<=row;t++) //loop that calculates the number at
each column in each row by using the nCr formula*/
        {
            printf(" %d", column_digit);

            column_digit=column_digit*numerator;

            column_digit=column_digit/denominator;

            numerator--;

            denominator++;

        }
        printf("\n");
    }
}
```

```
C:\Windows\SYSTEM32\cmd.exe
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1

-----
(program exited with code: 0)

Press any key to continue . . .
```

Problem 2:

Algorithm:

1. To create the columns headers, create one row with all column headers with indents
2. Use FILE function to open text file
3. Read the employee number, number of shifts, and wage
4. Repeat step 3 until there are no more values left in the text file
5. For each employee, add the total number of hours worked
6. And calculate gross pay by multiplying the total number of hours each employee worked by their wage
7. If the total hours are greater than 25, they get 10% extra income
8. If the total hours are in between 15 and 25, they get 5% extra income
9. Display all values in table

## Source code:

```
#include <stdio.h>

int main(void)
{
    int i, shifts, hours, total;

    double employee_num, wage, total_pay;

    FILE *text; /* opens text file with the employees' pay information*/
    text = fopen ("L4_data.txt", "r"); /* reads the file*/

    printf("Employee Number \t\tHours Worked\t\tTotal Pay \n"); //titles
    with indents to make output look like a table

    printf("-----\t\t\t-----\t\t\t-----\n");

    while (fscanf (text, "%lf %d %lf", &employee_num, &shifts, &wage) ==
3) // while loop runs until there are no more employee numbers or wages
to read//
    {
        total = 0;

        for (i=1; i<=shifts; ++i) // for loop inside the while loop
calculates the total pay based on the wages and the employees' hours
        {
            fscanf(text, "%d", &hours);

            total = total + hours;
        }

        if (total > 25){ // if statements based on the categories the
employees fall into
            wage = wage * 1.1;
        }

        else if (total > 15){
            wage = wage * 1.05;
        }

        total_pay = wage * total;

        printf("%.0lf \t\t\t\t%d\t\t\t$%.2lf \n", employee_num, total,
total_pay);
    }
}
```

```

    }

    fclose(text);

    return 0;
}

```

The screenshot shows a Windows command prompt window with the title bar 'C:\Windows\SYSTEM32\cmd.exe'. The window displays a table with three columns: 'Employee Number', 'Hours Worked', and 'Total Pay'. The data is as follows:

Employee Number	Hours Worked	Total Pay
77621	35	\$693.00
82010	16	\$378.00
92390	46	\$986.70
62396	12	\$384.00
89320	9	\$247.50
19089	43	\$756.80
54209	64	\$1196.80
50630	32	\$704.00

Below the table, the text '(program exited with code: 0)' is displayed, followed by 'Press any key to continue . . .'.

Problem 3:

Source code:

```

#include <stdio.h>

double pressure (double temp2)
{
    double pressure2, temp1, pressure1;

    temp1 = 300;

    pressure1 = 50;

    pressure2 = (pressure1*temp2) / temp1;
}

```

```

        return (pressure2);
    }

int main(void)
{
    double maxtemp, k, i;

    maxtemp = (300*500)/50; /*calculates the maximum temperature the
cylinder can hold*/

    printf("The tank can handle a temperature of %.2lf Kelvin before
exploding. \n", maxtemp);

    printf("Temperature (K)\t\t\tPressure (atm) \n");
    printf("-----\t\t\t----- ");

    for (i=273.5; i<maxtemp; i=i+100) /*Loop calculates the temperature
from 273.15 to maxtemp*/
    {
        k = pressure(i);

        printf("\n%.2lf ", i);

        printf(" \t\t\t%.2lf ", k);
    }

    printf("\n%.2lf ", i); /*displays final temperature and pressure*/
    printf(" \t\t\t%.2lf ", pressure(i));

    printf("\nKABOOM!");

    return 0;
}

```

```
C:\Windows\SYSTEM32\cmd.exe
The tank can handle a temperature of 3000.00 Kelvin before exploding.
Temperature (K)      Pressure (atm)
ΓÇö-----
273.50              45.58
373.50              62.25
473.50              78.92
573.50              95.58
673.50             112.25
773.50             128.92
873.50             145.58
973.50             162.25
1073.50            178.92
1173.50            195.58
1273.50            212.25
1373.50            228.92
1473.50            245.58
1573.50            262.25
1673.50            278.92
1773.50            295.58
1873.50            312.25
1973.50            328.92
2073.50            345.58
2173.50            362.25
2273.50            378.92
2373.50            395.58
2473.50            412.25
2573.50            428.92
2673.50            445.58
2773.50            462.25
2873.50            478.92
2973.50            495.58
3073.50            512.25
KABOOM!

-----
(program exited with code: 0)

Press any key to continue . . .
```