CPS 188 Lab 3 : Repetition & Loops

Instructor: *Dr. Ufkes*

TA: *Mohammed Emrul Hasan*

Section: *18*

Sayeed Ahamad

Student Number: *501209136*

1 Problem Sets

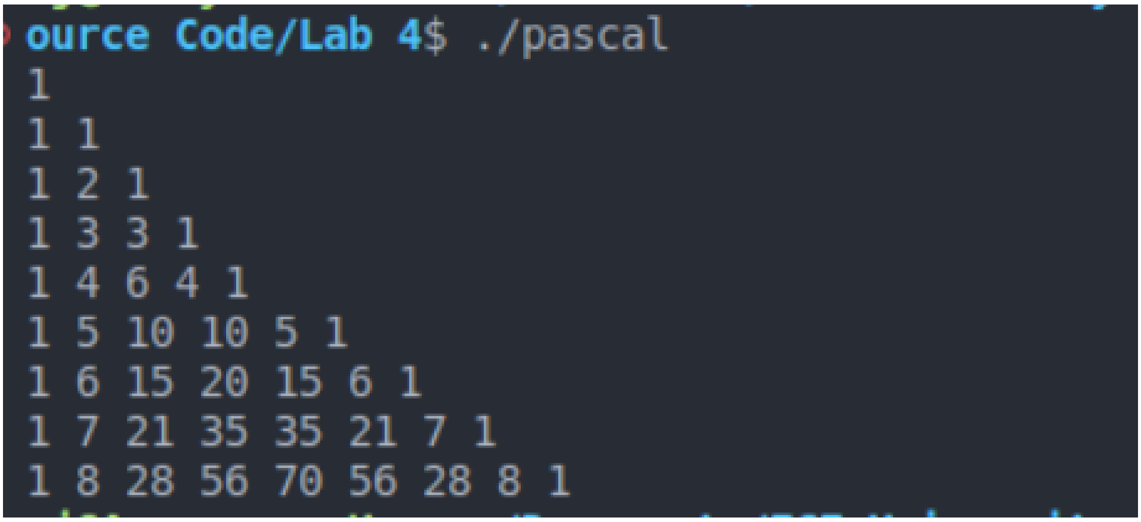
1.1 Problem 1

1.1.1 Computer Program

```c
/*  Program to print the values of the Pascal's Triangle in Sequential Order    */

#include <stdio.h>

void pascal(void);

void main(void)
{
    pascal();
}

void pascal(void)
{
    for (int i = 1; i <= 9; ++i)
       {

        int value = 1;

        for (int j = 1; j <= 9; ++j){

            if (value != 0){
            printf ("%d ", value);
            }

            value = value * (i-j)/j;

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

Listing 1.1: *Program to print the values of the Pascal's Triangle in Sequential Order*

1.1.2 Program Output Screenshot



```
ource Code/Lab 4$ ./pascal
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
```

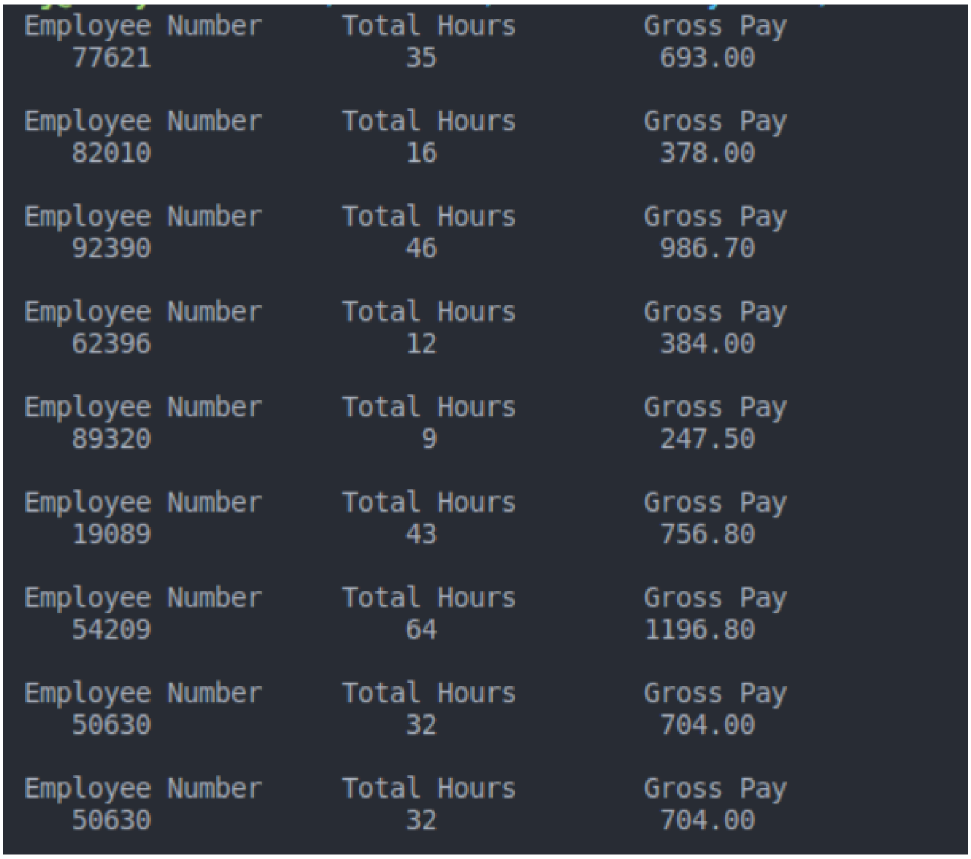
1.2 Problem 2

1.2.1 Computer Program

```c
/*  Program to calculate the Gross Pay of a series of workers */

#include <stdio.h>

int main()
{
    int employee_number, number_of_shifts, number_of_hrs, total_hrs,i;
    double wage_rate, gross_pay;
    FILE * in;
    in = fopen("L4_data.txt", "r");

    while(!feof(in)){
        fscanf(in, " %d %d %lf", &employee_number, &number_of_shifts, &wage_rate);
        i = 1;
        total_hrs = 0;
        while(i<= number_of_shifts){
            fscanf (in, "%d", &number_of_hrs);
            total_hrs = total_hrs + number_of_hrs;
            i++;
            }
            if (total_hrs <= 15){
                gross_pay = total_hrs * wage_rate;
                }
            else if (total_hrs >15 && total_hrs <= 25){
                gross_pay = (total_hrs * wage_rate * 1.05);
                }
            else if (total_hrs > 25){
                gross_pay = (total_hrs * wage_rate * 1.10);
                }
        printf("Employee Number     Total Hours        Gross Pay\n");
        printf("%8d%18d%20.2lf\n\n",employee_number,total_hrs,gross_pay);
    }

        fclose(in);
        return(0);
}


```

Listing 1.2: *Program to calculate the Gross Pay of a series of workers*

1.2.2 Program Output Screenshot



```
Employee Number     Total Hours        Gross Pay
   77621                 35              693.00

Employee Number     Total Hours        Gross Pay
   82010                 16              378.00

Employee Number     Total Hours        Gross Pay
   92390                 46              986.70

Employee Number     Total Hours        Gross Pay
   62396                 12              384.00

Employee Number     Total Hours        Gross Pay
   89320                  9              247.50

Employee Number     Total Hours        Gross Pay
   19089                 43              756.80

Employee Number     Total Hours        Gross Pay
   54209                 64             1196.80

Employee Number     Total Hours        Gross Pay
   50630                 32              704.00

Employee Number     Total Hours        Gross Pay
   50630                 32              704.00
```

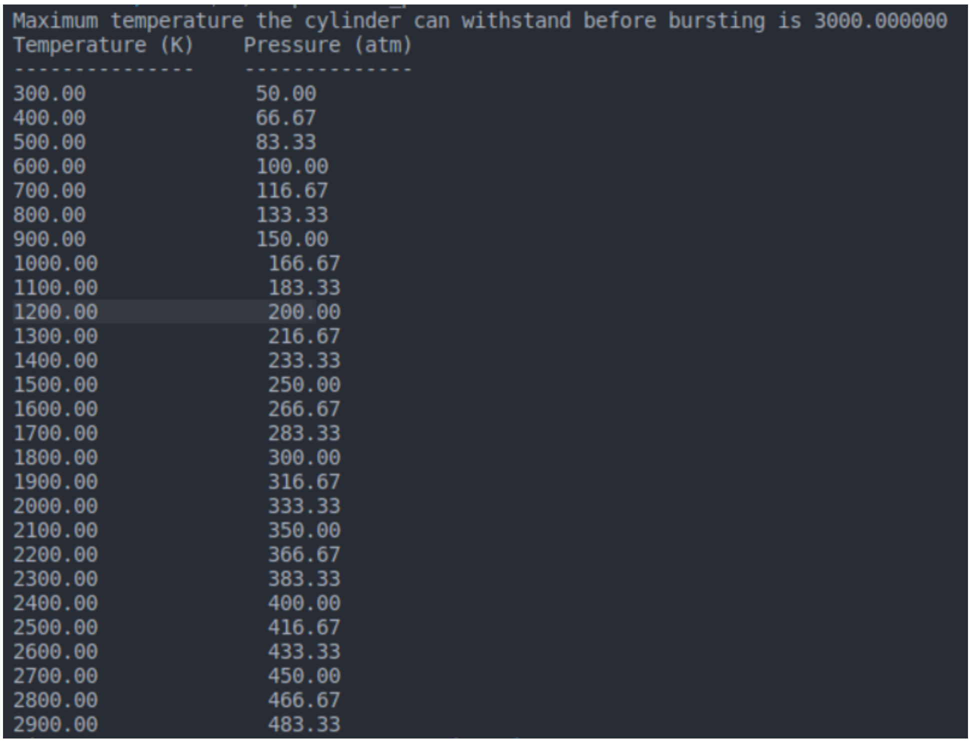
1.3 Problem 3

1.3.1 Computer Program

```c
/*  Program to Calculate the Temperature-Pressure relation for some Temperature and Pressure    */

#include <stdio.h>
#include <math.h>
#include <string.h>

float part_a(float initial_temperature, float initial_pressure, float final_pressure);
void part_b(float initial_temperature, float initial_pressure, float final_pressure, float max_temperature);

void main(void)
{
    float temp_i = 300;
    float pres_i = 50;
    float pres_f = 500;

    /*  Funtion to solve problem part a */
    float max_t = part_a(temp_i, pres_i, pres_f);

    /*  Funtion to solve problem part b */
    part_b(temp_i, pres_i, pres_f, max_t);
}

float part_a(float initial_temperature, float initial_pressure, float final_pressure)
{
    float max_temperature = (initial_temperature * final_pressure) / initial_pressure;
    printf("Maximum temperature the cylinder can withstand before bursting is %f\n", max_temperature);

    return max_temperature;
}

void part_b(float initial_temperature, float initial_pressure, float final_pressure, float max_temperature)
{
    int space = 4;

    /*  Header of the table */
    printf("Temperature (K)");
    for (int i = 0; i < space; i++)
    {
        printf(" ");
    }
    printf("Pressure (atm)\n");

    /*  Margins */
    for (int i = 0; i < strlen("Temperature (K)"); i++)
    {
        printf("-");
    }
    for (int i = 0; i < space; i++)
    {
        printf(" ");
    }
    for (int i = 0; i < strlen("Pressure (atm)"); i++)
    {
        printf("-");
    }
    printf("\n");

    /*  Contents of the table*/
    for (float temperature = initial_temperature; temperature < max_temperature; temperature += 100)
    {
        /*  Calculating iterative pressure*/
        float pressure = (initial_pressure * temperature) / initial_temperature;

        /*  Calculating & tabulating the temperature-pressure relation  */
        printf("%.2f", temperature);
        for (int i = 0; i < strlen("Temperature (K)") - 5 + space; i++)
        {
            printf(" ");
        }
        printf("%.2f", pressure);
        for (int i = 0; i < strlen("Pressure (atm)") - 4; i++)
        {
            printf(" ");
        }

        /*  Line termination print  */
        printf("\n");
    }
}


```

Listing 1.3: *Program to Calculate the Temperature-Pressure relation for some Temperature and Pressure*

1.3.2 Program Output Screenshot



```
Maximum temperature the cylinder can withstand before bursting is 3000.000000
Temperature (K)    Pressure (atm)
---------------    --------------
300.00              50.00
400.00              66.67
500.00              83.33
600.00              100.00
700.00              116.67
800.00              133.33
900.00              150.00
1000.00              166.67
1100.00              183.33
1200.00              200.00
1300.00              216.67
1400.00              233.33
1500.00              250.00
1600.00              266.67
1700.00              283.33
1800.00              300.00
1900.00              316.67
2000.00              333.33
2100.00              350.00
2200.00              366.67
2300.00              383.33
2400.00              400.00
2500.00              416.67
2600.00              433.33
2700.00              450.00
2800.00              466.67
2900.00              483.33
```