Pull with Git

```
/*
1.    Write a C Program for Simple Calculator using switch case statements
*/


#include <stdio.h>

int main()
{
    float a = 0 , b = 0;
    char response = '0';
    double result = 0;
    printf("Enter Two numbers: ");
    scanf("%f%f" , &a , &b );
    printf("What would you like to do with them\n1.) Add(a)\n2.) Subtract(s)\n3.)
Multiply(m)\n4.) Divide(d)\n");
    scanf(" %c" , &response);
    switch (response)
    {
    case 'a':
        result = a + b;
        break;
    case 's':
        result = a - b;
        break;
    case 'm':
        result = a * b;
        break;
    case 'd':
        result = a / b;
        break;
    default:
        printf("Wrong Response\n");
        return 0;
    }
    printf("Answer is: %0.2f" , result);
    return 0;
}



/*
2.    Write a C Program to print full Pyramid of numbers
      1
         2 3 2
      3 4 5 4 3
   4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
*/

#include <stdio.h>
int main()
{
   const int maxStage = 5;
   int stage = 1;
```

```
   int i = 1 , n = 0;
   for (n = 0; n < 2 * (maxStage - stage); n++)
      printf(" ");
   for (int i = 1; i <= maxStage * 2 - 1; i++)
   {
      if (i == stage * 2 - 1)
      {
         for (; i >= stage; i--)
            printf("%d ", i);
         printf("\n");
         i = stage * 2 - i;
         stage++;
         for (n = 0; n < 2 * (maxStage - stage); n++)
            printf(" ");
         if (stage > maxStage)
            break;
      }
      printf("%d ", i);
   }
}
```

```
/*
3.    Write a C program that takes a positive integer from the user and displays
all the positive factors of that number.
(Hint:
Enter a positive integer: 60
Factors of 60 are: 1 2 3 4 5 6 10 12 15 20 30 60)
*/

#include <stdio.h>
int main()
{
    int input;
    scanf("%d" , &input);
    for(int i = 1; i <= input; i++)
    {
        if(input%i == 0)
            printf("%d " , i);
    }
    return 0;
}
```

```
/*
4.    Write a C Program to count frequency of digits in a number
*/
#include <stdio.h>
#include <stdlib.h>
#define INTMAX 20

int main()
{
    char* buffer = (char*)malloc(INTMAX);
```

```
    int input; int i = 0;
    int digits[10] = {0};
    fgets(buffer , INTMAX, stdin );
    input = strtol(buffer , (char**)NULL , 10 );
    free(buffer);
    while(input != 0)
    {
        digits[input%10]++;
        input/=10;
    }
    for(i = 0 ; i < 10; i++)    {
        printf("%d  :  %d\n" , i , digits[i]);
    }
    return 0;
}



/*
5.    Write a C program to input number from user and check whether number is
Strong number or not.
*/
#include <stdio.h>
#include <stdlib.h>
#define INTMAX (int)__INT_MAX__
int factorial(int n)
{
    int result = 1;
    for (; n > 0; n--)
        result *= n;
    return result;
}
int sumFacdigs(int n)
{
    int sum = 0;
    while (n > 0)
    {
        sum += factorial(n % 10);
        n /= 10;
    }
    return sum;
}
int main()
{
    char *buffer = (char *)malloc(INTMAX);
    int input;
    fgets(buffer, INTMAX, stdin);
    input = strtol(buffer, (char **)NULL, 10);
    free(buffer);
    if (input == sumFacdigs(input))
    {
        printf("Power Number Found: %d", input);
        return 0;
    }
    printf("Not a Power Number: %d", input);
}
```