CS19101 PDS laboratory
Assignment 4

Write programs for problems 1,2 and 3 in three different files named A4_1_<machine number>_<Roll no.>.c, A4_2_<machine number>_<Roll no.>.c and A4_3_<machine number>_<Roll no.>.c respectively (without the '<' and '>'). Put these three files into a compressed directory named A4_<machine number>_<Roll no.>.zip and submit it.

Example: If your roll number is 19DEP99999 and your machine number is 99, then the names of your files should be A4_1_99_19DEP99999.c, A4_2_99_19DEP99999.c and A4_3_99_19DEP99999.c.

This is an assignment on functions. You are not allowed to use arrays. You are not allowed to use global variables. You are not allowed to change the prototype of the functions.

---

1. Write a function with the following prototype:
   int Isprime(int n);
   The above function takes an integer n as argument, and returns 1 if n is prime, and 0 otherwise. Now, in the **main()**, take two integers a and b as inputs through the keyboard. Assume that $a < b$. Use the function **Isprime()** to print all prime numbers that are strictly greater than a and strictly less than b, separated by comma's.

   Sample input/output:

   ```
   Enter a: 20
   Enter b: 55
   23,29,31,37,41,43,47,53
   ```

   [20 marks.]

2. Write a function with the following prototype:
   int factorial(int n);

that takes an integer argument and returns its factorial. Write a function with the following prototype:
int power(int n, int k);
that takes two integer arguments n and k, and returns n^k.
Now use these two functions to write another function with the following prototype:
double expon (int k, double d);
that takes an integer k and a double d as arguments, and returns an approximation of e^d by evaluating the following series upto the k-th term:

$$e^d = 1 + d + \frac{d^2}{2!} + \frac{d^3}{3!} + \dots$$

In **main()**, take an integer k and a double d as inputs through keyboard. Then compute the above approximation by calling function **expon**. Finally, print what **expon** returns on the screen.

[20 marks.]

3. Write a function with the following prototype:
void pdrom (int n);
that takes an integer as argument, and prints whether or not the input number is its own reverse (see sample input/output). In **main()**, take an integer n as input through keyboard; then pass n to the function **pdrom** to print whether or not n is its own reverse.

Sample input/output:

```
Enter integer: 100
No
Enter integer: 1331
yes
Enter integer: 3
yes
Enter integer: 40
no
```

[20 marks.]