Q1.

Read two integers from the user.

List and calculate sum of the numbers between these two numbers, recursively.

For example: User enters 10 and 12.

List = 10, 11 ,12

Sum = 10 + 11 + 12 = 33

Include a recursive function in your program (whose prototype is given below), which takes two numbers (where **`min <= max`**) and returns sum of the numbers between these two numbers, inclusively. This function should also print the numbers as shown in the sample run.

```
int sumOfRange(int min, int max);
```

Sample run:

```
Enter the first number:5
Enter the second number:15
Numbers are:
5 6 7 8 9 10 11 12 13 14 15
The sum of numbers from 5 to 15 : 110


Process returned 0 (0x0)   execution time : 20.060 s
Press any key to continue.
```

Q2.

Given a number n, print the following pattern without using any loop.

Examples :

Input: n = 16

Output: 16, 11, 6, 1, -4, 1, 6, 11, 16

Input: n = 10

Output: 10, 5, 0, 5, 10

We basically first reduce 5 one by one until we reach a negative or 0.

After we reach 0 or negative, we one add 5 until we reach n.

Include a recursive function (whose prototype is given below) in your program which takes an integer and prints the pattern mentioned above.

```
void print_pattern(int n);
```

Sample run:

```
Please enter an integer: 16
The pattern is: 16 11 6 1 -4 1 6 11 16
Process returned 0 (0x0)   execution time : 12.306 s
Press any key to continue.
```

Q3.

Write a recursive function (whose prototype is given below) which takes a non-negative integer N and returns the number of digits that N has.

```
int numberOfDigits(int n);
```

Sample run 1:

```
Enter an integer: 5876
5876 has 4 digits

Process returned 0 (0x0)   execution time : 3.735 s
Press any key to continue.
```

Sample run 2:

```
Enter an integer: 5
5 has 1 digits

Process returned 0 (0x0)   execution time : 2.643 s
Press any key to continue.
```

Q4.

Write a recursive function (whose prototype is given below) which takes two arguments: a non-negative integer N, and number of digits that N has. The function returns an integer in which the digits of N are reversed.

```
int reverseNumber(int n, int numOfDigits);
```

Note 1: You can use the function in Q3.

Note 2: You can use pow function from math.h library.

Sample run 1:

```
Enter an integer: 5876
5876 reversed: 6785

Process returned 0 (0x0)   execution time : 10.834 s
Press any key to continue.
```

Sample run 2:

```
Enter an integer: 5
5 reversed: 5

Process returned 0 (0x0)   execution time : 7.698 s
Press any key to continue.
```

Sample run 3:

```
Enter an integer: 24681
24681 reversed: 18642

Process returned 0 (0x0)   execution time : 22.305 s
Press any key to continue.
```