**Lab Assignment**

**CENG 2002 Programming Language Concepts, Spring 2021**

**Recursion**

* A recursive function is one that calls itself or that is part of a cycle in the sequence of function calls. The ability to invoke itself enables a recursive function to be repeated with different parameter values.
* Recursion is typically used to specify a natural, simple solution that would otherwise be very difficult to solve.
* One or more simple cases of the problem have a straightforward, nonrecursive solution. The other cases can be redefined in terms of problems that are closer to the simple cases.

**E.g. multiply(6,3)**

int multiply (int m, int n)

{

int ans;

if ( n == 1)

ans = m;

else

ans = m + multiply(m, n-1);

return (ans);

}

**Question 1**

Write a recursive function ***int* *nrDigits(int num)*** that counts the total number of digits of a number. Your program should read a positive integer value from the user, calls ***nrDigits*** function with this value as an argument and prints the result in the given format.

Sample Output:

Enter a positive integer number: 456654

Total digits in number 456654 is: 6

#include <stdio.h>

int nrDigits (int num)

{

static int count=0;

if(num>0)

{

count++;

nrDigits (num/10);

}

else

{

return count;

}

}

int main(void)

{

int number;

int count=0;

printf("Enter a positive integer number: ");

scanf("%d",&number);

count=nrDigits(number);

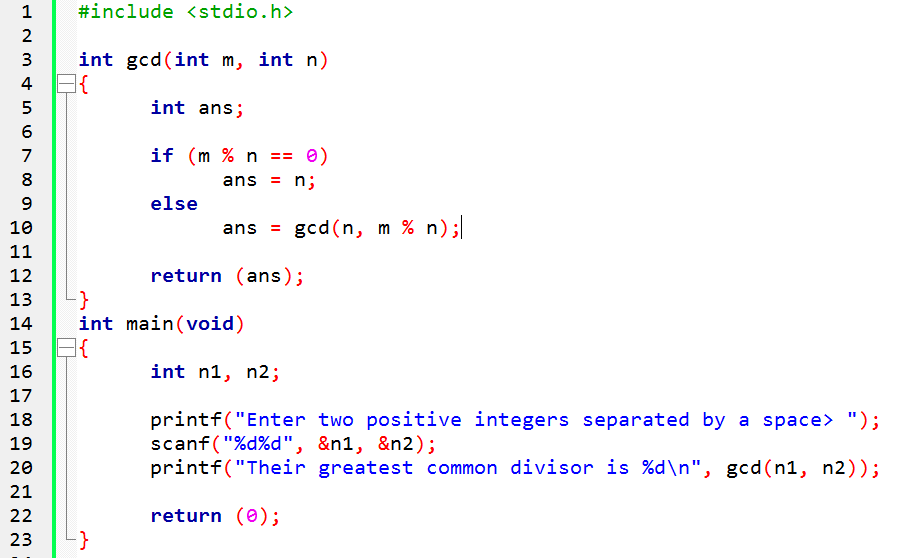
printf("Total digits in number %d is: %d\n",number,count);

return 0;

}

**Question 2**

Write GCD function in a recursive way. Greatest common divisor of two integers is the largest integer that divides them both evenly.



**Question 3**

Write a recursive function which checks whether a given number is prime or not.

1. #include <stdio.h>
3. int primeno(int, int);
5. int main()
6. {
7. int num, check;
8. printf("Enter a number: ");
9. scanf("%d", &num);
10. check = primeno(num, num / 2);
11. if (check == 1)
12. {
13. printf("%d is a prime number**\n**", num);
14. }
15. else
16. {
17. printf("%d is not a prime number**\n**", num);
18. }
19. return 0;
20. }
22. int primeno(int num, int i)
23. {
24. if (i == 1)
25. {
26. return 1;
27. }
28. else
29. {
30. if (num % i == 0)
31. {
32. return 0;
33. }
34. else
35. {
36. return primeno(num, i - 1);
37. }
38. }
39. }

HOME EXERCISE

Remember Tower of Hanoi and write the function recursively.

The Tower of Hanoi is a mathematical game or puzzle. It consists of three rods and a number of disks of different sizes, which can slide onto any rod

The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

* Only one disk can be moved at a time.
* Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or on an empty rod.
* No larger disk may be placed on top of a smaller disk.
* Move n disks from Peg A to peg C using peg B as needed.
* The following conditions apply:
  + Only one disk at a time may be moved, and this disk must be the top disk on a peg.
  + A larger disk can never be placed on top of a smaller disk.
  + The minimum number of moves required to solve a Tower of Hanoi puzzle is 2*n*− 1, where *n* is the number of disks.

