Offline

Chapter -8,17

General Instructions

1. In case of recursive procedure, you can't use any data register to store the intermediate results, but you can use the registers for intermediate calculation. You have to store the intermediate result into the stack.
2. You have to convert the input digits into an integer input and no input size will be greater than 16 bit.
3. Print all the outputs (except strings) as decimal number and no output size will be greater than 16 bit.
4. *Please do not copy*

OFFLINE FOR A1

A. Write a recursive procedure`gen_number(a,b)` that generates a number following the recursive formula written in C below:

```
int gen_number(int a,int b)
{
    if(a<=1) return 2;

    if(b<=2) return 3;

    return gen_number(a-1,b)+gen_number(a,b-1);


}
```

B. Write a procedure `get_unit(n)`that returns the unit's digit of a number. For example, if n is 13 it returns 3.

C. Take two input a and b. Now using above two procedures generate a number and as output print the number and whether its unit digit is 0 or not.

Input	Output
a: 5 b: 4	40 Yes
a: 4 b: 3	11 No

OFFLINE FOR A2

A. Write a recursive procedure `gen_series(n)` that generates the `nth` term of a series that follows the following recurrence relation:

$S_n = S_{n-1} + S_{n-2} + S_{n-3} - 5$
Where $S_0 = 3$, $S_1 = 6$, $S_2 = 4$

Sample Implementation in C:

```c
int gen_series(int n)
{
    if(n==0) return 3;
    if(n==1) return 6;
    if(n==2) return 4;
    return gen_series(n-1)+gen_series(n-2)+gen_series(n-3)-5;
}
```

B. Write a procedure `sum(n)` that finds the sum of first n term of a series.

C. Now using the above two procedures find the sum of first n terms of the aforementioned series where n is given as input by user.

Input	Output
n: 5	34
N: 9	269

OFFLINE FOR B1

A. Write a recursive procedure `base_convert(number, base)` which converts **number** which is in decimal format into desired **base (2<=base<=10)**.

Sample Implementation in C:

```c
int base_convert(int number,int base){
    if(number == 0 || base==10)
        return number;

    return (number % base) + 10*base_convert(number / base, base);
}
```

B. Write a procedure that `count_nonzero (n)` counts the number of non-zero digits in a number n.

C. Now take input a decimal number n. Now using the above two procedures print the number of 1 bits in its binary representations.

Input	Output
5	2
100	3

OFFLINE FOR B2

A. Write a recursive procedure `fibonacci(n)` which calculates the `n'th` fibonacci number.
B. Write a procedure `isPrime(n)` which check if a given number n is prime or not.
C. Now using these functions determine the first n fibonacci numbers that are also prime where n is given as input by user.

Input	Output
5	2 3 5 13 89
7	2 3 5 13 89 233 1597