```
#include <stdio.h>
#include <stdlib.h>



/**
 *  This is the simple fibonacci sequence generator.
 *  First few numbers of series are 0, 1, 1, 2, 3, 5, 8 etc, Except
 *  first two terms in sequence every other term is the sum of two
 *  previous terms, For example 8 = 3 + 5 (addition of 3, 5)
 */
fibonacci(int n)
{
        int a = 0;
        int b = 1;
        int i;

        /**
        *    * Here is the standard for loop. This will
        *    step through, performing the code
        *    * inside the braces until i is equal to n.
        *    */
        for (i=0;i<=n;i++)
        {
                int sum;
                sum = a+b;
                a = sum;
                b = a;
                printf(“%d”, sum);
        }
        return 0;
}

int main()
{
        int *n;

        while(1)
        {
                /* printf prints a formated string to the stdout */

                printf("\nHow many numbers of the sequence would you like?\n");

        /* scanf reads a formated string from the stdin */
        n = (int *) malloc(sizeof(int));
        scanf("%d",&n);
```

```
        /* break the loop when n is 0 */
        if( n = 0 )
            break;

        /* Here we call the fibonacci function */
        fibonacci(n);
    }

}
```

2) Explain the significance of the colon (:) in the structure defined below

```
struct str
        {
                int i;
                unsigned a1:4;
                unsigned a2:2;
                unsigned a3:6;
        };
```

3) Swap odd bits with even bits in an integer

Swap 0 & 1, 2 & 3 etc

```
Int swap(int input){
        // Int num_of_bit = sizeof(input)*8;
        Int odd_bit = input && 0b10101010….10;
        Int even_bit = input && 0b0101010….01;
        //Bool leftmost = input && (1<<num_of_bit);
        //Bool rightmost = input && 1;
        Return odd_bit<<1 | even_bit>>1;
}
```

Add 1 to a "string" of digits.

"5" "6" "7" + 1 = "5" "6" "8"

```
Int AddOne(string input){
        Int addon = 1;
        for(auto it = input.end()-1; it!=input.start(); --it){
                &it = str(int(*it) + addon);
                addon = 0;
                if( *it == ‘0’’){
                        addon = 1;
                }
        }
}
```

Return to_string(int(input)+1)