Homework- Recursion

1. Predict the output of the following program. What does the following fun() do in general?

```
#include <stdio.h>

int fun ( int n, int *fp )
{
    int t, f;

    if ( n <= 1 )
    {
        *fp = 1;
        return 1;
    }
    t = fun ( n-1, fp );
    f = t + *fp;
    *fp = t;
    return f;
}

int main()
{
    int x = 15;
    printf("%d\n",fun(5, &x));

    return 0;
}
```

2.

Consider the following recursive C function. Let *len* be the length of the string s and *num* be the number of characters printed on the screen, give the relation between *num* and *len* where *len* is always greater than 0.

```
void abc(char *s)
{
    if(s[0] == '\0')
        return;

    abc(s + 1);
    abc(s + 1);
    printf("%c", s[0]);
}
```

3.Explain the functionality of following functions

```
int fun1(int x, int y)
{
  if(x == 0)
    return y;
  else
    return fun1(x - 1,  x + y);
}
```

4.Explain the functionality of following functions.

```
* Assume that n is greater than or equal to 1 */
int fun1(int n)
{
  if(n == 1)
     return 0;
  else
     return 1 + fun1(n/2);
}
```

```
5./* Assume that n is greater than or equal to 0 */
void fun2(int n)
{
  if(n == 0)
    return;

  fun2(n/2);
  printf("%d", n%2);
}
```

6.Explain the functionality of below recursive function.

```
void fun1(int n)
{
   int i = 0;
   if (n > 1)
     fun1(n-1);
   for (i = 0; i < n; i++)
     printf(" * ");
}
```

7.

Explain the functionality of below recursive function.

```
#define LIMIT 1000
void fun2(int n)
{
  if(n > LIMIT)
    return;
  printf("%d ", n);
  fun2(2*n);
  printf("%d ", n);
}
```

8. Predict the output of following program.

```
#include<stdio.h>
void fun(int x)
{
  if(x > 0)
  {
     fun(--x);
     printf("%d\t", x);
     fun(--x);
  }
}

int main()
{
  int a = 4;
  fun(a);
  getchar();
  return 0;
}
```

9.Predict the output of following program. What does the following fun() do in general?

```
int fun(int a[],int n)
{
  int x;
  if(n == 1)
    return a[0];
  else
    x = fun(a, n-1);
  if(x > a[n-1])
    return x;
  else
    return a[n-1];
```

```
}

int main()
{
  int arr[] = {12, 10, 30, 50, 100};
  printf(" %d ", fun(arr, 5));
  getchar();
  return 0;
}
```

10. Reverse a stack using recursion

You are not allowed to use loop constructs like while, for..etc, and you can only use the following ADT functions on Stack S:
isEmpty(S)
push(S)
pop(S)