## Exercises-Ch7

1.We assume that the storage organization and the form of activation record used in C language program run-time stack storage allocation are as following. Please **construct the run-time stack map when it gets the maximum size for the first time** for the following C program (10%).

## Storage Organization of C Language

Activation Record of the function called by Main function
Activation Record of Main function
Global Variable Data Area

The Form of Activation Record of any a function in C

Unit for Returned Value of function	TOP	<b>†</b>
Internal Temporary Work Units	101	
Local Variable Data Storage Units	-	
Formal parameter Storage Units	-	
Number of Formal Parameters	1	
Returned Address	1	
Caller's SP (Start Address of caller's activation record)	→ SP	

```
The C program is as the following:
#include <stdio.h>
int x,y;

int main()
{
```

```
x=5;
  y=f(x);
}
int f(int n)
{
  if (n<=1)
       return 1;
 else
       if ( n==2)
             return 2;
       else
        {
       int t1,t2,t3,t4,t;
       t1=f(n-1);
       t2=f(n-2);
       t3=f(n-3);
       t4=t1+t2
          t=t3+t4;
       return t
        }
```

Notes: 1) Here we assume that the caller's sp of Main function is the start address of global variable data area, and the returned address in the

activation record of a function (including Main function) is filled by the operating system automatically, you might not care it.

- 2) The initial value of variable X is 5, the start address of stack used in the program is K.
- 3) The stack map may get its maximum size for several times, here we ask you draw the stack map at maximum size for the first time.
- 2. We assume that the storage organization and the form of activation record used in C language program run-time stack storage allocation are as above. Please **construct the run-time stack map when it gets the maximum size** for the following C program (10%).

```
The C program is as the following:
#include <stdio.h>
int x,y,z;
 int main()
   x=40;
   y=35;
   z=f(x,y);
 int f(int m, int n)
   if (n>m) {
    int t;
    t=f(n,m);
    return t }
   else
     if (n==0) return m;
     else {
       int t1.t2:
        t1=m % n; //remainder of m/n
       t2=f(n,t1);
       return t2
        }
```

}

**Notes:** 1) Here we assume that the caller's sp of Main function is the start address of global variable data area, and the returned address in the activation record of a function (including Main function) is filled by the operating system automatically, you might not care it.

2) The initial value of variable X and Y are 40 and 35 respectively, the start address of stack used in the program is K.