

数据结构 作业2

王华强 2016K8009929035

第3章:

3-3

写出下列程序段的输出结果:

```
void main()
{
    Stack S;
    char x, y;
    InitStack(S);
    x = 'c'; y = 'k';
    Push(S, x);   Push(S, 'a'); Push(S, y);   Pop(S, x);
    Push(S, 't'); Push(S, x);   Pop(S, x);   Push(S, 's');
    while(!StackEmpty(S))
    {
        Pop(S,y);
        printf(y);
    }
    printf(x);
}
```

//stack

3-7

按照四则运算加、减、乘、除和幂运算(t)优先关系的惯例，并仿照教科书3.2节例3-2的格式，画出对下列算术表达式求值时操作数栈和运算符栈的变化过程：

A-BxC/D+E↑F

很容易根据函数运行结果判断pop,push函数对应的栈, 因此这里在函数调用中不再显式指出操作的栈.

EXPRESSION	FUNC_CALL	OPERATOR	NUMBER
A-BxC/D+E↑F	init	#	-
-BxC/D+E↑F	push(A)	#	A
BxC/D+E↑F	push(-)	#-	A
xC/D+E↑F	push(B)	#-	AB
C/D+E↑F	push(*)	#-*	AB
/D+E↑F	push(C)	#-*	ABC
/D+E↑F	pop(),pop(),P=B*C,push(P)	#-	AP
D+E↑F	push(/)	#-/	AP
+E↑F	push(D)	#-/	APD

+E↑F	L=P/D	#-	AL
+E↑F	M=A-L	#	M
E↑F	push(+)	#+	M
↑F	push(E)	#+	ME
F	push(↑)	#+↑	ME
-	push(F)	#+↑	MEF
-	(meet #),K=E↑F	#+	MK
-	(meet #),J=M+K	#	J
-	(meet #),返回结果J	#	J

3-10 试将下列递归过程改写为非递归过程。

```
void test(int &sum)
//use cpp
{
    int x;
    cin>>x;
    if(x==0)
        sum=0;
    else
    {
        test(sum);
        sum+=x;
    }
    cout<<sum;
}
```

改写:

```
void test(int &sum)
{
    int x;
    SetupStack(C);

    do{
        scanf("%d",&x)
        push(x,C);
    }while(x!=0);

    sum=0;

    while(StackNotEmpty(C))
    {
        printf("%d",sum);
        pop(C,x);
        sum+=x;
    }

    printf("%d",sum);
}
```