DS Homework 5

注：请使用 A4 纸作答，写上姓名学号，并于下一次上课时提交。

1、利用两个栈 s1 和 s2 模拟一个队列时，如何用栈的运算来实现该队列的以下运算，请用类 C/C++语言写出伪代码。

函数：

enqueue: 插入一个元素　　dequeue: 删除一个元素

queue_empty: 判定队列为空

```
void enqueue (SqStack &s1, SElemType x)
{    push(s1, x);    }

void dequeue (SqStack &s1, SElemType &x)
{    SqStack s2;
     SElemType y;
     s2.top = s2.base;
     while (!sempty(s1))
     {    pop(s1, y);    push(s2, y);    }
     pop(s2, x);
     s1.top = s1.base;
     while (!sempty(s2))
     {    pop(s2, y);    push(s1, y);    }
}

Status queue_empty (SqStack s1)
{    return sempty(s1);  }
```

2、已知 Ackerman 函数的定义如下：

$$akm(m,n)=\begin{cases} n+1 & m=0 \\ akm(m-1,1) & m\neq 0, n=0 \\ akm(m-1,akm(m,n-1)) & m\neq 0, n\neq 0 \end{cases}$$

(1) 写出递归算法

(2) 利用栈操作写出非递归算法

(1) 递归算法

```
int akm(int m, int n)
{     if (m==0) return (n+1);
      else if (n==0) return akm(m-1, 1);
      else return akm(m-1, akm(m, n-1));
}
```

(2) 非递归算法

```
constant max = 100;
typedef struct {
      int mval; int nval;
} stack;

int akm(int m, int n)
{     // S[max]为附设栈，top 为栈顶指针
      stack S[max];
      int top=0;

      S[top].mval = m; S[top].nval = n;
      do {
            while(S[top].mval)
            {     while(S[top].ncal)
                  {     top++; S[top].mval = S[top-1].mval;
                        S[top].nval = S[top-1].nval - 1;
                  }
                  S[top].mval--; S[top].nval=1;
            }
            if (top>0)
            {     top--; S[top].mval--;
                  S[top].nval = S[top+1].nval + 1;
            }
      } while (top !=1 || S[top].mval != 0);
      top--;
      return (S[top+1].nval+1);
}
```