## 栈和队列练习题

1. 设输入序列1、2、3、4，则下述序列中（ ）不可能是出栈序列

A. 1、2、3、4                      B. 4、 3、2、1

C. 1、3、4、2                      D.４、1、2、3

2. 若入栈序列是 a, b, c, d, e，则不可能的出栈序列是（ ）。

（A） edcba（B）decba（C）dceab （D）abcde

3、判定一个栈 ST(最多元素为m0) 为空的条件是（ ）。  
    （A） ST.top != ST.base         （B）ST.top == ST.base  
  （C）ST.top != ST.base +m0   （D）ST.top == ST.base+ m0

3、判定一个栈 ST(最多元素为m0) 为满的条件是（ ）。  
 （A） ST.top != ST.base         （B）ST.top == ST.base  
 （C）ST.top != ST.base +m0   （D）ST.top == ST.base+ m0

4. 栈和队列的主要区别在于（）

A. 它们的逻辑结构不一样 B. 它们的存储结构不一样 C. 所包含的元素不一样 D. 插入、删除操作的限定不一样

5. 循环队列存储在数组A[0…n]，则入队时的操作为（ ）

A. rear=rear+1

B. rear=(rear+1)mod(n-1)

C. rear=(rear+1)mod n

D. rear=(rear+1)mod(n+1)

6. 若用数组A[0…5]来实现循环队列，且当前rear和front的值分别为1和5，当从队列中删除一个元素，再加入两个元素后，rear和front的值分别为（）

A. 3和4

B. 3和0

C. 5和0

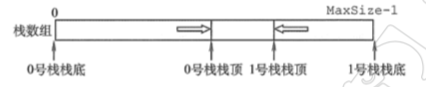
D. 5和1

7. 已知循环队列存储在一维数组A[0…n-1]，且队列非空时，front和rear分别指向队头元素和队尾元素。若初始时队列为空，且要求第一个进入队列的元素存储在A[0]处，则初始时front和rear的值分别是（）

A. 0,0 B. 0，n-1 C. n-1,0 D. n-1,n-1

8 设栈S和队列Q的初始状态为空，元素a、b、c、d、e、f依次通过栈S，一个元素出栈后即进入队列Q，若这6个元素出队列的顺序是b、d、c、f、e、a则栈S的容量至少应该是多少？ （3分）3

9. 如图 若栈采用顺序存储方式存储，现两栈共享空间V[0…MaxSize-1]，top[i]代表第i个栈( i =0,1)栈顶，栈1的底在v[0]，栈2的底在V[MaxSize-1]，则这两个栈满的条件是什么? (4分)



### top【0】+top[1]=maxsize;

10. 请设计一种栈式结构，要求实现Push(入栈)、Pop(出栈)、Min(返回栈中的最小值)三种操作。 并用5, 6, 9, 2 这四个数据辅助说明这个栈的设计，及其出栈，入栈和取最小值的操作。

#include<bits/stdc++.h>

#define maxstack 100

using namespace std;

enum Error\_code {

underflow,overflow,success

};

class Stack{

public:

Stack();

bool Empty()

{

return count==0;

}

Error\_code pop();

Error\_code push(int &n);

Error\_code min(int &m);

private:

int count;

int a[maxstack];

};

Stack::Stack()

{

count=0;

memset(a,0,sizeof(a));

}

Error\_code Stack::pop()

{

if(Empty())

{

return underflow;

}

count--;

return success;

}

Error\_code Stack::push(int &n)

{

if(count>=maxstack)

{

return overflow;

}

a[count]=n;

if(count>0&&n>a[count-1])

{

pop();

}

count++;

return success;

}

Error\_code Stack::min(int &m)

{

if(Empty())

{

return underflow;

}

m=a[count-1];

return success;

}

int main()

{

Stack stack;

char ch;

int N,i;

cin>>N;

int a[1000];

for(i=0;i<N;i++)

{

cin>>a[i];

stack.push(a[i]);

}

int m;

stack.min(m);

cout<<m;

return 0;

}