#include <iostream>

#include<time.h>

using namespace std;

struct Stack1 {

int top;

int\* data;

};

void InitStack(Stack1& st, int capacity) {

st.data = new int[capacity];

st.top = -1;

}

void push(Stack1& st, int value) {

st.data[++st.top] = value;

}

int pop(Stack1& st) {

return st.data[st.top--];

}

void nullStack(Stack1& st) {

st.top = -1;

}

bool empty(Stack1& st) {

return st.top == -1;

}

void PrintStack(Stack1& st) {

for (int i = st.top; i >= 0; i--)

cout << st.data[i] << endl;

cout << endl;

}

struct Queue1 {

int head, tail, size;

int\* data;

};

void nullQueue(Queue1& q) {

q.head = 0; q.tail = q.size - 1;

}

void InitQueue(Queue1& q,int capacity) {

q.size = capacity + 1;

q.data = new int[q.size];

nullQueue(q);

}

int next(Queue1& q, int n) {

return (n + 1) % q.size;

}

bool empty(Queue1& q) {

return next(q, q.tail) == q.head;

}

void add(Queue1& q, int value) {

if (next(q, next(q, q.tail)) == q.head)

cout << "Queue overflow" << endl;

else {

q.tail = next(q, q.tail);

q.data[q.tail] = value;

}

}

int del(Queue1 & q) {

if (empty(q)) {

cout << "Queue is empty" << endl; return 0;

}

else {

int d = q.data[q.head];

q.head = next(q, q.head);

return d;

}

}

int main(){

srand(time(NULL));

Stack1 st1, st2;

int capacity=1+ rand()%20;

InitStack(st1, capacity);

for (int i=0; i <= capacity; i++) {

push(st1, rand() % 1000);

}

capacity =1+rand() % 20;

InitStack(st2, capacity);

for (int i=0; i <= capacity; i++) {

push(st2, rand() % 1000);

}

PrintStack(st1);

PrintStack(st2);

Queue1 Que1;

capacity = 1 + rand() % 20;

InitQueue(Que1, capacity) {

}

system("pause");

return 0;

}