栈队列源代码：

#include<stdio.h> // EOF(=^Z或F6),NULL

#include<stdlib.h> // srand( ) ,rand( ),exit(n)

#include<malloc.h> // malloc( ),alloc( ),realloc( )等

#include<limits.h> // INT\_MAX等

#include<string.h>

#include<ctype.h>

#include<math.h> // floor(),ceil( ),abs( )

#include<iostream> // cout,cinusing namespace std;

#include<time.h> // clock( )

#define TRUE 1

#define FALSE 0

#define OK 1

#define ERROR 0

#define INFEASIBLE -1

#define OVERFLOW -2

typedef int Status;

typedef int QElemType;

#define MAXSIZE 100 //最大长度

typedef struct QNode // 结点类型

{

QElemType data;

struct QNode \*next;

} QNode, \*QueuePtr;

typedef struct

// 链队列类型

{

QueuePtr front; //队头指针

QueuePtr rear; //队尾指针

} LinkQueue;

//1.初始化带头结点的空队列

Status InitQueue (LinkQueue &Q)

{

Q.front=Q.rear=new QNode;

if(!Q.front) exit(OVERFLOW);

Q.front->next=NULL;

return OK;

}

//3.入队

Status EnQueue(LinkQueue &Q,QElemType &e)

{

QNode \*p= new QNode;

p->data=e;

p->next=NULL;Q.rear->next=p;

Q.rear=p; //修改队尾指针

return OK;

}

//2.创建

Status CreatQueue(LinkQueue &Q,int &n)

{

QNode \*p = new QNode; // 创建一个新节点

Q.front = Q.rear = p; // 初始化队头和队尾指针

p->next = nullptr; // 新节点的下一个节点为空

int e;

printf( "请输入队列的长度:");

scanf("%d",&n); // 输入队列的长度

for (int i = 0; i < n; i++)

{

printf( "请输入要输入的数: " ) ;

scanf("%d",&e); // 输入队列的元素

p = new QNode; // 创建一个新的节点

p->data = e; // 设置节点的数据

p->next = nullptr; // 新节点的下一个节点为空

Q.rear->next = p; // 将新节点链接到队尾

Q.rear = p; // 更新队尾指针

}

return 0; // 假设返回0表示成功

}

//4.删除队头元素，并删除后输出

Status DeQueue(LinkQueue &Q,QElemType &e)

{

QNode \*p= new QNode;

if(Q.front==Q.rear) return ERROR;

p=Q.front->next;

e=p->data;

Q.front->next=p->next;

if(Q.rear==p) Q.rear=Q.front;

delete p;

return OK;

}

//5.输出队头元素

Status GetHead (LinkQueue Q, QElemType &e)

{

if(Q.front==Q.rear) return ERROR;

e=Q.front->next->data;

return OK;

}

6.输出队列中所有元素

Status print\_Q(LinkQueue &Q)

{

if(Q.front==Q.rear) return ERROR;

QNode \*p=Q.front->next;

printf("队列中的元素为：");

while(p!=NULL)

{

printf("%d ",p->data);

p=p->next;

}

printf("\n");

return OK;

}

int main()

{

LinkQueue Q;

QElemType x;

int e,n,choice;

while(1)

{

printf("\n主菜单如下：\n");

printf("1.初始化一个带头结点的空队列\n");

printf("2.根据随机输入的队列长度和队列中整数型数据元素的值，创建一个非空链队列\n");

printf("3.将一个数x插到队尾（x在程序运行时随机输入）\n");

printf("4.删除队头元素，并在删除结束后输出删除元素的值\n");

printf("5.判定队列是否为空，若不空输出队头元素\n");

printf("6.输出队列中所有元素\n");

printf("0.退出\n");

printf("请输入你的选择（0-6）：");

scanf("%d",&choice);

switch(choice)

{

case 1:

InitQueue(Q);

printf("初始化成功!\n");

break;

case 2:

CreatQueue(Q,n);

print\_Q(Q);

break;

case 3:

printf("插入的数为:");

scanf("%d",&e);

EnQueue(Q,e);

print\_Q(Q);

break;

case 4:

if(DeQueue(Q,e)==OK)

{

printf("删除的元素值为：%d\n",e);

print\_Q(Q);

}

else

printf("队列为空!\n");

break;

case 5:

if(GetHead(Q,e)==OK)

{

printf("队头元素为：%d\n",e);

}

else

printf("队为空！\n");

break;

case 6:

print\_Q(Q);

break;

case 0:

free(Q.front);

break;

default:

printf("无效选择，请重新输入！\n");

}

}

}