链栈：

代码：

（截图在代码后面）

#include <iostream>

using namespace std;

typedef struct

{

int num;

} SElemType;

typedef struct StackNode

{

SElemType data;

struct StackNode \*next;

} StackNode, \*LinkStack;

void menu()

{

cout << "=================================" << endl;

cout << "1.压入栈顶" << endl;

cout << "2.查看栈顶元素" << endl;

cout << "3.取栈顶元素" << endl;

cout << "4.遍历链栈栈" << endl;

cout << "5.置空链栈" << endl;

cout << "6.数制转换操作" << endl;

cout << "7.退出" << endl;

cout << "=================================" << endl;

}

void Input(LinkStack &L) // 插入链栈

{

StackNode \*q, \*p = L;

q = new StackNode;

cin >> q->data.num;

L = q;

q->next = p;

}

void Check\_front(LinkStack L) // 输出栈顶

{

cout << "栈顶元素是：" << L->data.num << endl;

}

void Get\_front(LinkStack &L, SElemType &nm) // 取栈顶元素

{

nm.num = L->data.num;

L = L->next;

}

void Show\_stack(LinkStack L) // 遍历链栈

{

while (L->next != NULL)

{

cout << L->data.num << " ";

L = L->next;

}

cout << endl;

}

void Empty\_stack(LinkStack &L) // 置空链栈

{

StackNode \*p;

while (L->next != NULL)

{

p = L;

L = L->next;

delete p;

}

}

void Ten\_to\_eight(LinkStack &L, int k)//数制转换

{

while(k >= 8)

{

StackNode \*q, \*p = L;

q = new StackNode;

q->data.num = k % 8;

L = q;

q->next = p;

k /= 8;

}

StackNode \*q, \*p = L;

q = new StackNode;

q->data.num = k;

L = q;

q->next = p;

while (L->next != NULL)

{

cout << L->data.num;

L = L->next;

}

cout << endl;

}

int main()

{

StackNode \*L;

L = new StackNode;

L->data.num = 0;

L->next = NULL;

while (1)

{

menu();

char option;

cout << "请选择：";

cin >> option;

fflush(stdin);

if(option < '1' || option > '8')

{

cout << "输入不合法" << endl;

continue;

}

switch (option)

{

case '1':

int m;

cout << "请输入需要插入元素个数：";

cin >> m;

cout << "请输入需要插入的元素" << endl;

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

{

Input(L);

}

cout << "插入成功" << endl;

break;

case '2':

Check\_front(L);

break;

case '3':

SElemType s;

Get\_front(L, s);

cout << "栈顶元素是：" << s.num << endl;

break;

case '4':

cout << "从栈顶到栈底依次为：";

Show\_stack(L);

break;

case '5':

Empty\_stack(L);

cout << "置空成功" << endl;

break;

case '6':

int k;

cout << "请输入要转换的数（十进制转八进制）：";

cin >> k;

cout << "转换后为：";

Empty\_stack(L);

Ten\_to\_eight(L, k);

break;

case '7':

cout<<"退出成功"<<endl;

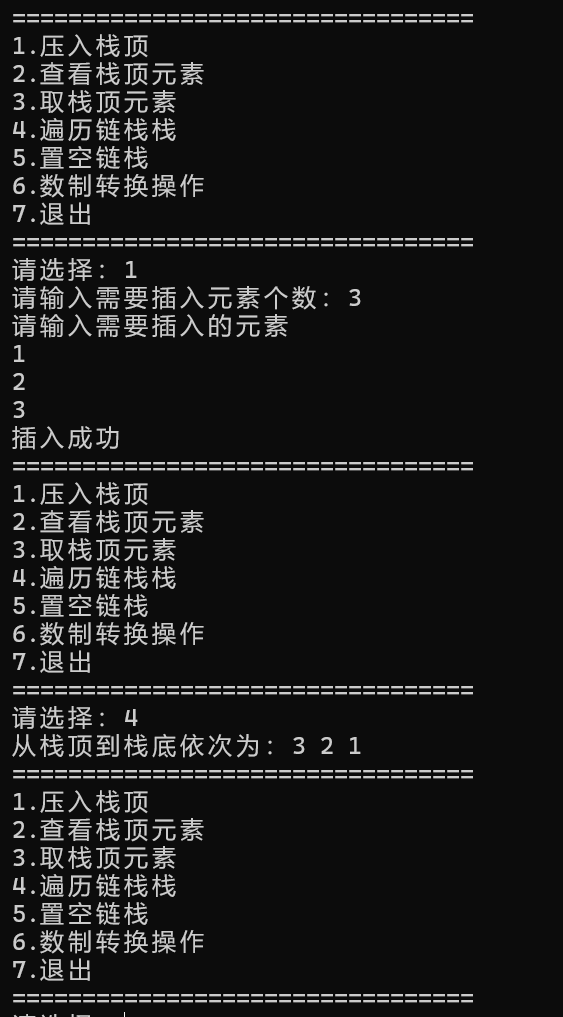
exit(1);

}

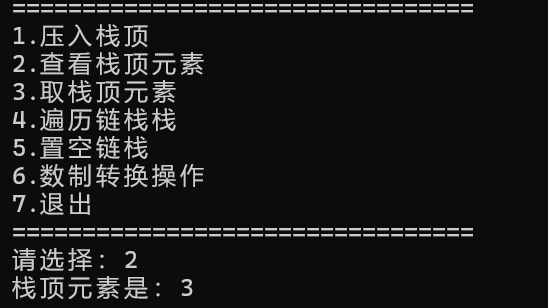
}

}

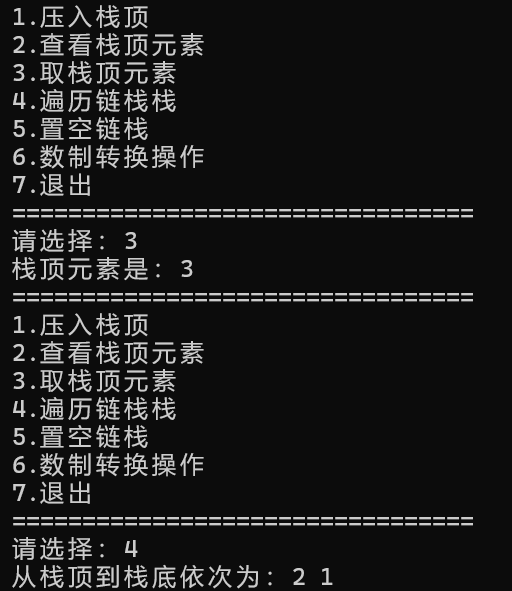
压入栈顶



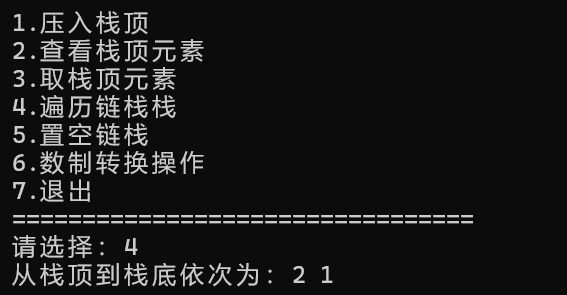
查看栈顶元素：



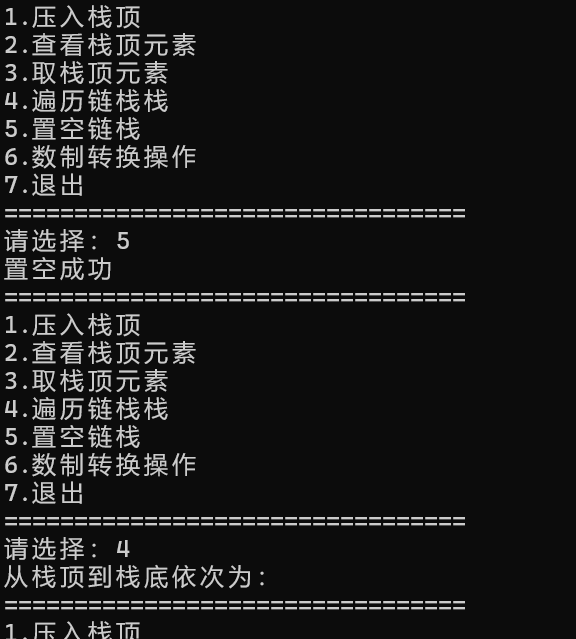
取栈顶元素：



遍历链栈：



置空链栈：



数制转换：

