1，约瑟夫环：

#include<iostream>

using namespace std;

int main()

{

int a[1000],b[1000],n,m,i,k=0,j=0;

cout<<"请输入参数："<<endl;

cin>>n>>m;

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

a[i]=i+1;

while(j!=n)

{

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

{

if(a[i]==0) continue;

k++;

if(k!=m) continue;

k = 0;

b[j++]=a[i];

a[i]=0;

}

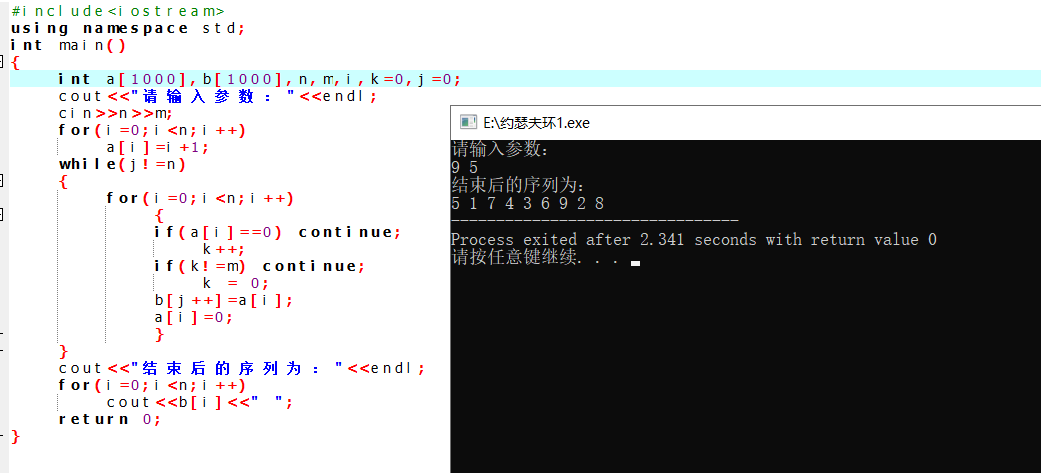
}

cout<<"结束后的序列为："<<endl;

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

cout<<b[i]<<" ";

return 0;

}

2，单链表逆序

#include<iostream>

#include<stack>

#include<assert.h>

using namespace std;

typedef struct node{

int data;

node \* next;

}node;

node \* add(int n, node \* head){

node \* t = new node;

t->data = n;

t->next = NULL;

if (head == NULL){

head = t;

}

else if (head->next == NULL){

head->next = t;

}

else{

node \* p = head->next;

while (p->next != NULL){

p = p->next;

}

p->next = t;

}

return head;

}

void print(node \* head){

node \* p = head;

while (p != NULL){

cout << p->data << " ";

p = p->next;

}

cout << endl;

}

void reversePrint(node \* p){

if (p != NULL){

reversePrint(p->next);

cout << p->data << " ";

}

}

void reversePrint2(node \* head){

stack<int> s;

while (head != NULL){

s.push(head->data);

head = head->next;

}

while (!s.empty()){

cout << s.top() << " ";

s.pop();

}

}

int main(){

node \* head = NULL;

for (int i = 1; i <= 8; i++){

head = add(i, head);

}

print(head);

reversePrint(head);

reversePrint2(head);

system("pause");

return 0;

}