#### 1. 用C/C++将一组输入的整型数据保存在一个单项链表中，并实现其插入和删除操作。

#include<iostream>

#include<string>

#include <stdio.h>

using namespace std;

struct Node {

int content;

Node \*next;

};

Node \*creat()

{

Node \*head, \*p;

head = new Node;

p = head;

int x, cycle = 1;

while (cycle)

{

cin >> x;

if (x != -1)

{

Node \*s = new Node;

s->content = x;

p->next = s;

p = s;

}

else

{

cycle = 0;

}

}

head = head->next;

p->next = NULL;

return head;

}

bool insert(Node \*&h, int a, int pos)

{

Node \*p = new Node;

p->content = a;

int i = 1;

int tag = 0;

if (h == NULL)

{

h = p;

p->next = NULL;

tag = 1;

}

else

{

Node \*pn = h;

Node \*qn = h;

while (pn->next != NULL)

{

pn = pn->next;

i++;

}

if (pos == 0)

{

p->next = h;

h = p;

tag = 1;

}

else

{

if (pos >= i)

{

pn->next = p;

p->next = NULL;

tag = 1;

}

else

{

while (pos>1)

{

qn = qn->next;

pos--;

}

p->next = qn->next;

qn->next = p;

tag = 1;

}

}

}

if (tag == 1)

return true;

else

return false;

}

bool remove(Node \*&h, int &a, int pos)

{

Node \*p = h;

Node \*q1 = h;

Node \*before = h;

Node \*after = NULL;

int tag = 0;

int i = 1;

if (h == NULL)

i = 0;

else

{

while (q1->next != NULL)

{

q1 = q1->next;

i++;

}

}

if (i>0)

{

if (pos >= i)

{

while (before->next != NULL)

{

after = before;

before = before->next;

}

if (after != NULL)

{

a = before->content;

after->next = NULL;

delete before;

i = i - 1;

tag = 1;

}

else

{

a = before->content;

h = NULL;

delete before;

i = i - 1;

tag = 1;

}

}

else

{

if (pos == 1)

{

a = p->content;

h = h->next;

delete p;

i = i - 1;

tag = 1;

}

else

{

while (pos>1)

{

after = before;

before = before->next;

pos--;

}

a = before->content;

after->next = before->next;

delete before;

i = i - 1;

tag = 1;

}

}

}

else

cout << "链表为空!" << endl;

if (tag == 1)

return true;

else

return false;

}

void output(Node \*h) {

for (Node \*p = h; p != NULL; p = p->next)cout << p->content << ',';

cout << endl;

}

int main()

{

int num;

int pos = 0;

bool t;

int i = 0;

Node \*h = creat();

output(h);

cout << "输入插入的值：" << endl;

cin >> num;

cout << "输入插入的结点：" << endl;

cin >> pos;

insert(h, num, pos);

output(h);

cout << "输入删除的结点：" << endl;

cin >> pos;

remove(h, num, pos);

cout << "删除的值为：" << num << endl;

output(h);

}

**2. 编写一个函数，将任意输入的字符串按字母序顺序输出。**

**输入示例：dafceaabd**

**输出示例：aaabcddef**

***使用库函数***

#include<iostream>

#include<cstdlib>

using namespace std;

int cmp(const void \*a, const void \*b)

{

return \*((char\*)a) - \*((char\*)b);

}

int main()

{

char s[500] = { 0 };

while (cin >> s)

{

qsort(s, strlen(s), sizeof(char), cmp);

cout << s << endl;

}

}

***不使用库函数***

#include<iostream>

#include<string>

using namespace std;

void sort(char \*a) {

int i, j, k, len, temp;

for (len = 0; \*(a + len); len++);

for (i = 0; i<len - 1; i++)

{

k = i;

for (j = i + 1; j<len; j++)

if (\*(a + j)<\*(a + k))k = j;

temp = \*(a + i);

\*(a + i) = \*(a + k);

\*(a + k) = temp;

}

}

int main()

{

char a[200];

cin >> a;

sort(a);

cout << a << endl;

return 0;

}