# 标准冒泡排序

#include<iostream>

using namespace std;

void main()

{

int a[] = { 6,5,9,7,2,8 };

int n = sizeof(a) / sizeof(int \*);

int temp;

for (int i = n - 1; i > 0; i--)

{

int flag = 0;

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

if (a[j]>a[j + 1])

{

flag = 1;

temp = a[j];

a[j] = a[j + 1];

a[j + 1] = temp;

}

if (flag == 1)

{

cout << "第" << n - i << "趟：";

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

cout << a[k] << " ";

cout << endl;

}

}

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

cout << a[i] << " ";

cout << endl;

system("pause");

}

# 链表实现

#include<iostream>

using namespace std;

struct list{

int num;

list \*next;

};

list \*head=NULL;

void creat(int n)

{

while (n--)

{

int temp;

cin >> temp;

list \*newnode= new list;

newnode->num = temp;

newnode->next = head;

head = newnode;

}

}

void display()

{

list \*p = head;

while (p != NULL)

{

cout << p->num << " ";

p = p->next;

}

cout << endl;

}

void sortp(int n)

{

for (int i = n - 1; i > 0;i--)

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

{

list \*ptr1 = head, \*ptr2;

for (int k = 0; k < j; k++)

ptr1 = ptr1->next;

ptr2 = ptr1->next;

if (ptr1->num > ptr2->num)

{

int temp;

temp=ptr1->num;

ptr1->num = ptr2->num;

ptr2->num = temp;

}

}

}

int main()

{

cout << "排序个数：" << endl;

int n;

cin >> n;

creat(n);

display();

sortp(n);

display();

system("pause");

return 0;

}

# 选择排序

#include<iostream>

using namespace std;

void display(int a[],int n)

{

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

{

cout << a[i] << " ";

}

cout << endl;

}

void main()

{

int a[] = { 6,5,9,7,1,8 };

int n = sizeof(a) / sizeof(int \*);

int temp,k=1;

for (int i = 0; i < n - 1; i++)

{

int min = a[i],pos=i,temp,flag=0;

for(int j = i+1; j < n; j++)

if (min>a[j])

{

flag = 1;

min = a[j];

pos = j;

}

if (flag == 1)

{

temp = a[pos];

a[pos] = a[i];

a[i] = temp;

cout << "第" << k++<< "趟:";

display(a, n);

}

}

cout << "排序后结果：";

display(a, n);

system("pause");

}

# 插入排序

#include<iostream>

#include<iomanip>

using namespace std;

#define N 5//待排序元素个数

//输出数组

void Display(int \*data, int n)

{

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

cout << setw(3) << data[i];

cout << endl;

}

//插入排序

void InsertSort(int \*data, int n)

{

for (int i = 1; i < n; i++)

{

int temp = data[i],j=i-1;

while (j >= 0 && data[j]>temp)

{

data[j + 1] = data[j];

j--;

}

data[j + 1] = temp;

cout << "第" << i << "趟：";

Display(data, N);

}

}

//主函数

void main()

{

int data[N] = { 6, 4, 9, 8, 3 };

cout << "排序前的数据为: ";

Display(data, N);

InsertSort(data, N);//插入排序

cout << "排序后的数据为: ";

Display(data, N);

system("pause");

}

# 希尔排序

#include<iostream>

#include<iomanip>

using namespace std;

#define N 8//待排序元素个数

//输出数组

void Display(int \*data, int n)

{

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

cout << setw(3) << data[i];

cout << endl;

}

//希尔排序

void ShellSort(int \*data, int n)

{

int i;//i为扫描次数

int j;//以j来定位比较的元素

int k = 1;//k输出计数

int temp;//temp用来暂存数据

int jump;//设定间距步长

jump = n / 2;

while (jump != 0)

{

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

{

temp = data[i];

j = i - jump;

while (j >= 0 && temp<data[j])//插入排序法

{

data[j + jump] = data[j];

j = j - jump;

}

data[jump + j] = temp;

}

cout << "第" << k++ << "次排序:\t";

Display(data, n);

jump = jump / 2;//控制循环数

}

}

//主函数

void main()

{

int data[N] = { 6, 9, 2, 3, 4, 7, 5, 1 };

cout << "排序前的数据为: ";

Display(data, N);

ShellSort(data, N);//希尔排序

cout << "排序后的数据为: ";

Display(data, N);

system("pause");

}