（1）题目描述：使用动态方式生成数组p对其排序，输出排序后的结果。在排序过程中使用两种策略实现：

1）改变数组p中的元素顺序（交换元素值），

2）不改变数组p中的元素顺序（交换元素地址）。

（2）思路：即将所需排序的数组分为有序和无序两个部分且有序区在前。每次从无序区中取出一个元素插到有序区的适当位置。

如果需要排序的元素插入有序区的中间，则在插入元素后有序区的每一位元素后移。

1. 代码实现：

1）

void InsertSort(int \*p,int n){

int temp; //用来存储下一个要排序的元素

int sorted; //有序区的长度

int i,j;

for(i = 1 ; i < n ; i++){//从第二个元素开始排序

sorted = i-1; //有序区从头到将要排序的元素的前一个

temp = p[i];

while(sorted >= 0 && p[sorted] > temp){ //从后向前比较，将比较和后移工作放在一个while循环里，因为不清楚循环次数所以使用while循环

p[sorted+1] = p[sorted]; //从要排序的元素开始，前一个元素向后移

sorted--;

}

p[sorted+1] = temp; //while循环结束即要排序的元素应在的位置

}

}

int main(){

int n,i;

scanf("%d",&n);

int \*p = (int\*)malloc(sizeof(int) \* n);

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

scanf("%d",&p[i]);

}

InsertSort(p,n);

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

printf("%d",p[i]);

}

}

2）

#include <stdio.h>

#include <stdlib.h>

void InsertSort(int \*p,int n){

int temp;

int sorted;

int i,j;

int \*s = p,\*e = p+n-1;

for( ; s < e ; s++){

sorted = s-p-1;

temp = \*s;

while(sorted >= 0 && \*(p+sorted) > temp){

\*(p+sorted+1) = \*(p+sorted);

sorted--;

}

\*(p+sorted+1) = temp;

}

}

int main(){

int n,i;

scanf("%d",&n);

int \*p = (int\*)malloc(sizeof(int) \* n);

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

scanf("%d",&p[i]);

}

InsertSort(p,n);

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

printf("%d",p[i]);

}

}