实验一

编程题 2 （总分：100.00）

#	题目	分值
1	**实验1-1: Permutation with Repetition**	50.00

Description

R={ r1,r2,⋯ ,rn }是要进行排列的n 个元素。其中元素r1,r2,⋯ ,rn可能相同。试设计一个算法，列出R的所有不同排列。
编程任务：给定n 以及待排列的n 个元素。计算出这n 个元素的所有不同排列。

Input

输入由多组测试数据组成。每组测试数据的第1 行是元素个数n，1 <= n <= 500。接下来的1 行是待排列的n 个元素。

Output

对应每组输入，将计算出的n 个元素的所有不同排列输出，每种排列单独一行。最后1 行中的数是排列总数。

Sample Input

4

aacc

Sample Output

aacc

acac

acca

caac

caca

ccaa

6

参考答案:

```
#include <stdio.h>
#include <algorithm>
using   namespace std ;
int ans ;
int ok(char str[],int a ,int b )
{
        if( b > a)
         for(int i = a ; i< b ; i++)
                 if( str[i] == str[b] )
                         return 0 ;
         return 1 ;
}
void perm(char str[],int k ,int m)
{
         int i ;
         if( k == m )
         {
                 ans ++ ;
                 for( i = 0 ;i <= m ;i++ )
                 {
                         printf("%c",str[i] ) ;
                 }
                 printf("\n") ;
         }
         else
         {
              for( i = k ; i <= m ;i++)
                 if( ok(str,k,i) )
                 {
                         swap ( str[k],str[i] );
                         perm(str, k+1 , m );
                         swap(str[k],str[i] ) ;
                 }
         }
}
int main(int argc, char* argv[])
{
   char str[1000];
         int n ;
         while( scanf("%d",&n) != EOF )
   {
    ans = 0 ;
    scanf("%s",str ) ;
    perm(str,0,n-1) ;
    printf("%d\n",ans );
   }
         return 0;
}
```

2 实验1-2: Search Number 50.00

Description

科研调查时得到了n个自然数，每个数均不超过1500000000。已知不相同的数不超过10000个，现在需要在其中查找某个自然数，如找到则输出并统计这个自然数出现的次数，如没找到则输出NO。

Input

输入由多组测试数据组成。
每组测试数据输入包含n+1行；
第一行是两个整数n和x，n表示自然数的个数,x表示要查找的自然数，两者之间用空格隔开；
第2至n+1每行一个自然数。

Output

对应每组输入，如果查找到x，则每行输出两个整数，分别是自然数和该数出现的次数，其间用一个空格隔开；如果没有查找到x，则每行输出NO.

Sample Input

8 100

2

4

2

4

5

100

2

100

Sample Output

100 2

参考答案:

```
#include<stdio.h>
#include<string.h>
#define LEN 200000
int a[LEN],temp,mid;
int sort(int *a,int low,int high) //一趟快排
{
```

```
mid=a[low];
while (low<high)
{
while (low<high && a[high]>=mid) high--;
temp=a[low];a[low]=a[high];a[high]=temp;
while (low<high && a[high]<=mid) low++;
temp=a[low];a[low]=a[high];a[high]=temp;
}
return low;
}
void quicksort(int *a,int low,int high) //快排递归
{
//int mid;
if (low < high)
{
mid=sort(a,low,high);
quicksort(a,low,mid-1);
quicksort(a,mid+1,high);
}
}
int main()
{
int i,n,s;
int Sum=0;
scanf("%d",&n);
scanf("%d",&s);
for (i=0;i<n;i++)
{scanf("%d",&a[i]);}
quicksort(a,0,n-1); //调用快排
for (i=0;i<n;i++) //统计不同数字的个数
{
        if(a[i]==s)
        {
                Sum++;
        }
}
if(Sum==0)
{
        printf("NO");
}
else
{
    printf("%d %d",s,Sum);
}
}
```