

上机实验题

1 读入一组整数 ($0 < n < 100$)，将该组数按从小到大排序后，输出其中重复出现的每一个数（输入中一定有重复的数）。要求：设计 sort 函数用于排序，设计 output 函数用于输出重复出现的数值，在 main 函数中调用 sort 和 output。

输入样例：第一行为数的个数，第二行为具体的数

10

3 5 6 10 1 3 6 6 20 1

输出样例：按从小到大输出重复出现的数

1 1 3 3 6 6 6

```
1 #include <stdio.h>
2 void swap(int *a, int *b)/*交换两个数的值*/{
3     int t = *a;
4     *a = *b;
5     *b = t;
6 }
7 void sort(int *a, int n)/*这里采用冒泡排序*/{
8     for (int i = 0; i < n; i++)
9         for (int j = 0; j < n - 1; j++) {
10             if (a[j] > a[j + 1])
11                 swap(&a[j], &a[j + 1]);/*如果当前数组元素大于后面那个 则换位置，达到一个排序效果*/
12         }
13 }
14 void output(int *a, int n) {
15     for (int i = 0; i < n; i++)
16     {
17         if (a[i] == a[i + 1] || a[i] == a[i - 1]) // 如果当前数组元素和前或后一个元素相同则说明重复
18         {
19             printf("%d ", a[i]);/* 输出当前重复的数组元素*/
20         }
21     }
22     printf("\n");
23 }
24 int main() {
25     int n;
26     scanf("%d", &n);/* 读入数组大小*/
27     int a[n];
28     for (int i = 0; i < n; i++) /*读入数组*/
29         scanf("%d", &a[i]);
30     sort(a, n);/*调用sort函数*/
31     output(a, n);/*调用output函数*/
32     return 0;
33 }
```

10
3 5 6 10 1 3 6 6 20 1
1 1 3 3 6 6 6

Run Succeeded Time 39 ms Peak Memory 8.1M

2 输出以下的杨辉三角形，要求输出 10 行，设计函数 `gen` 用于生成杨辉三角形，在 `main` 函数中输出。

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

The screenshot shows a C programming IDE with a file explorer on the left, a code editor in the center, and an output window at the bottom. The code defines a function `gen` to generate the Pascal's triangle and prints it in the `main` function. The output window displays the first 10 rows of the triangle.

```
#include <stdio.h>
#define N 10
void gen (int a,int i,int j); //定义gen函数
int main (void)
{
    int i,j,a[N][N];
    for (i=0;i<N;i++)
    {
        a[i][i]=1;
        a[i][0]=1;
    }
    for (i=2;i<N;i++)
    {
        for (j=1;j<=i-1;j++)
            a[i][j]=a[i-1][j-1]+a[i-1][j];
    }
    for (i=0;i<N;i++)
    {
        for (j=0;j<=i;j++)
            printf("%6d",a[i][j]);
        printf("\n");
    }
    printf("\n");
    return 0;
}
```

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
```

Run Succeeded Time 34 ms Peak Memory 8.0M

3 求组合数 $c(m, n)$, or \square_n^m , 其中: $20 > m, n > 0$ 。

输入样例: 一行里两个数, 代表 m 和 n

20 10

输出样例: 一个数, 代表求出的数值

184756

The screenshot shows a C code editor with a file explorer on the left and a code editor on the right. The file explorer lists various C files, including 'c-projects', 'include', 'char', 'getchar', 'tempCodeRunnerFile', 'while', '作业1', '作业1.c', '单链表.c', '寻找素数.c', '循环练习.c', '指针1.c', '搜索.c', '数组1.c', '普通的运算符加减乘除运算样板.c', '比较三个数最大值.c', '练习1.c', '练习加法.c', '计算时间差.c', '输入金额练习.c', '递归练习.c', '陈伟杰71066001第一次作业第...', '陈伟杰71066001第一次作业第...', '顺序表 动态分配和静态分配.c', and '顺序表结构.c'. The code editor shows a C program that calculates the factorial of a number. The program includes `<stdio.h>`, defines `int main()`, and declares `int n, m;`. It uses `double` for `n1`, `m1`, and `o1`. The program reads input `n` and `m` using `scanf`, then calculates the factorial of `n` using a `for` loop. The output is printed using `printf`. The output window shows the input `20 10` and the output `184756`. The status bar at the bottom indicates 'Run Succeeded', 'Time 33 ms', and 'Peak Memory 7.8M'.

```
1 #include<stdio.h>
2 int main()
3 {
4     int n,m;
5     double n1,m1,o1;
6     double fact(int n);
7     printf("\n");
8     scanf("%d%d",&n,&m);
9     n1=fact(n);
10    m1=fact(m);
11    o1=fact(n-m);
12    printf("%0.0f\n",n1/(m1*o1));
13 }
14 double fact(int n)//求阶乘
15 {
16
17     ...int i;
18     ...double sum;
19     sum=1;
20     for(i=1;i<=n;i++)
21     {
22         sum=sum*i;
23     }
24     return sum;
25 }
```

20 10
184756

Run Succeeded Time 33 ms Peak Memory 7.8M