

打印九九乘法表

```
1 //由于不懂是哪个版本的所以我都写了
2 #include<stdio.h>
3 int main() {
4     //正常版本
5     for (int i = 1; i <= 9; i++) {          //控制行
6         for (int j = 1; j <= 9; j++) {      //控制列
7             if (j == 1) {
8                 printf("%d*%d=%2d", j, i, i * j);
9             }
10            else {
11                printf("%2d*%d=%2d", i, j, i * j);
12            }
13        }
14        printf("\n");
15    }
16    //倒三角版本
17    //for (int i = 1; i <= 9; i++) {          //控制行
18    //    for (int j = 9; j >= i; j--) { //控制列
19    //        if (j == 9) {
20    //            printf("%d*%d=%d", (10 - j), i, i * (10 - j));
21    //        }
22    //        else {
23    //            printf("%2d*%d=%d", (10 - j), i, i * (10 - j));
24    //        }
25    //    }
26    //    printf("\n");
27    //}
28    //正三角版本
29    //for (int i = 1; i <= 9; i++) {          //控制行
30    //    for (int j = 1; j <= i; j++) { //控制列
31    //        if (j == 1) {
32    //            printf("%d*%d=%2d", j, i, i * j);
33    //        }
34    //        else {
35    //            printf("%2d*%d=%2d", j, i, i * j);
36    //        }
37    //    }
38    //    printf("\n");
39    //}
40    return 0;
41 }
```

```
1*1= 1 1*2= 2 1*3= 3 1*4= 4 1*5= 5 1*6= 6 1*7= 7 1*8= 8 1*9= 9
1*2= 2 2*2= 4 2*3= 6 2*4= 8 2*5=10 2*6=12 2*7=14 2*8=16 2*9=18
1*3= 3 3*2= 6 3*3= 9 3*4=12 3*5=15 3*6=18 3*7=21 3*8=24 3*9=27
1*4= 4 4*2= 8 4*3=12 4*4=16 4*5=20 4*6=24 4*7=28 4*8=32 4*9=36
1*5= 5 5*2=10 5*3=15 5*4=20 5*5=25 5*6=30 5*7=35 5*8=40 5*9=45
1*6= 6 6*2=12 6*3=18 6*4=24 6*5=30 6*6=36 6*7=42 6*8=48 6*9=54
1*7= 7 7*2=14 7*3=21 7*4=28 7*5=35 7*6=42 7*7=49 7*8=56 7*9=63
1*8= 8 8*2=16 8*3=24 8*4=32 8*5=40 8*6=48 8*7=56 8*8=64 8*9=72
1*9= 9 9*2=18 9*3=27 9*4=36 9*5=45 9*6=54 9*7=63 9*8=72 9*9=81
```

E:\王道\基础测试\day01 陈瑞松\Debug\1. 打印九九乘法表.exe (进程 3332)已退出，代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口。...

2.a打印菱形

```
1  #include<cstdio>                                //分成上下两部分打印
2  #include<iostream>
3  using namespace std;
4  int main() {
5      for (int i = 1; i <= 5; i++) {                //打印上半部分（5行）
6          for (int j = 0; j <= 4 - i; j++) {         //上半部分空格打印 4行 4-3-2-1
7              cout << " ";
8          }
9          for (int j = 1; j <= 2 * i - 1; j++) {      // 上半部分 * 打印 5行 1-3-5-
10             7-9
11             if (j % 2 == 0) {
12                 cout << " ";                       //偶数个挖空
13             }
14             else {
15                 cout << "*";
16             }
17             cout << endl;
18         }
19         for (int i = 4; i > 0; i--) {
20             for (int j = 1; j <= 5 - i; j++) {       //下半部分空格打印 4行 1-2-3-4
21                 cout << " ";
22             }
23             for (int j = 1; j <= 2 * i - 1; j++) {    // 下半部分 * 打印 4行 7-5-3-
24                 2-1
25                 if (j % 2 == 0) {
26                     cout << " ";                       //偶数个挖空
27                 }
28                 else {
29                     cout << "*";
30                 }
31                 cout << endl;
32             }
33             return 0;
34         }
```

```

*
* *
* * *
* * * *
* * * * *
* * * * *
* * * *
* * *
* *
*

```

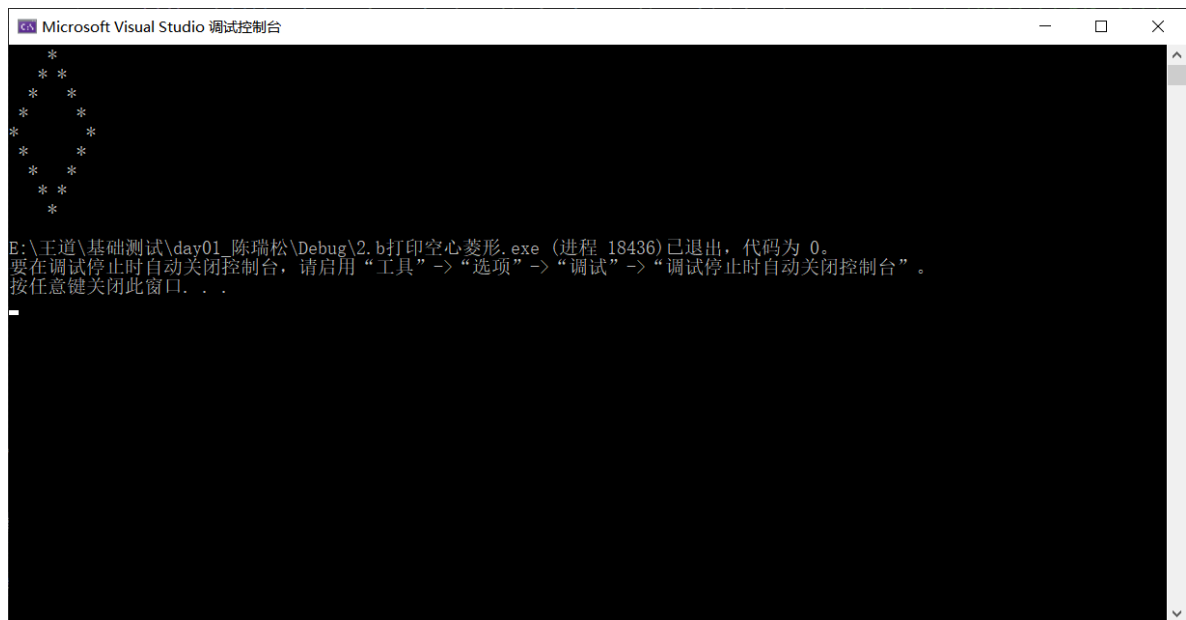
E:\王道\基础测试\day01_陈瑞松\Debug\2. a打印菱形.exe (进程 13416) 已退出，代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口。...

2.b 打印空心菱形

```

1  #include<cstdio>                                //分成上下两部分打印
2  #include<iostream>                                //可以看作是大菱形套了一个小的由"
   "组成的菱形
3  using namespace std;
4  int main() {
5      for (int i = 1; i <= 5; i++) {                //打印上半部分 (5行)
6          for (int j = 0; j <= 4 - i; j++) {         // 上半部分空格打印 (4行 4-3-2-
   1)
7              cout << " ";
8          }
9          cout << "*";                                //先打印一条边
10         for (int j = 1; j <= 2 * i - 2; j++) {      //再打印剩下部分 (4行 2-4-6-8)
11             if (j == 2 * i - 2) {                  //仅在末尾输出""
12                 cout << "*";
13             }
14             else {
15                 cout << " ";
16             }
17         }
18         cout << endl;
19     }
20     for (int i = 4; i > 0; i--) {
21         for (int j = 1; j <= 5 - i; j++) {          //下半部分空格打印 4行 1-2-
   3-4
22             cout << " ";
23         }
24         cout << "*";
25         for (int j = 1; j <= 2 * i - 2; j++) {      //下半部分打印 3行 6-4-2
26             if (j == 2 * i - 2) {
27                 cout << "*";
28             }
29             else {
30                 cout << " ";
31             }
32         }
33         cout << endl;
34     }
35     return 0;
36 }

```



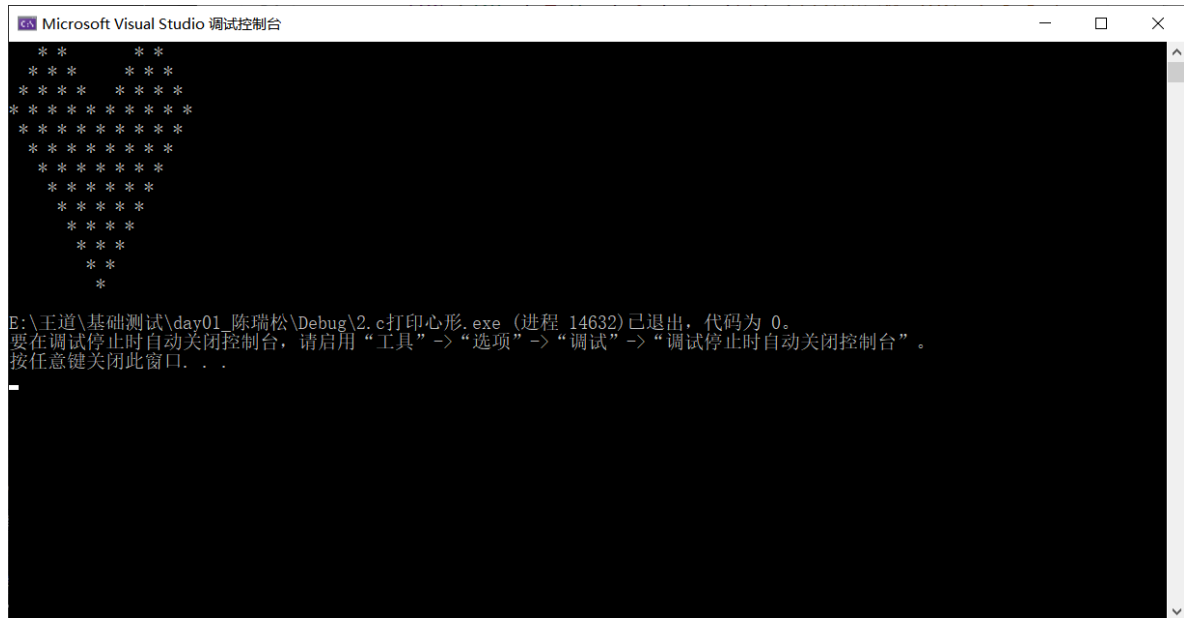
2.c 打印心型

```
1  #include<iostream> //分部分打印
2  #include<cstdio>
3  using namespace std;
4  int main() {
5      for (int i = 1; i <= 3; i++) { //打印前三行
6          for (int j = 0; j <= 3 - i; j++) { //空格打印 2行 2-1
7              cout << " ";
8          }
9          for (int j = 1; j <= 2 * i + 1; j++) { //"*"打印 3行 3-5-7 偶数挖空
10             if (j % 2 == 0) {
11                 cout << " ";
12             }
13             else {
14                 cout << "*";
15             }
16         }
17         for (int j = 1; j <= 9 - 2 * i; j++) {
18             cout << " ";
19         }
20         for (int j = 1; j <= 2 * i + 1; j++) {
21             if (j % 2 == 0) {
22                 cout << " ";
23             }
24             else {
25                 cout << "*";
26             }
27         }
28         cout << endl;
29     }
30     for (int i = 1; i <= 10; i++) { //打印下半部分 10行
31         for (int j = 0; j < i - 1; j++) { //打印空格 10行 1-3-5-7-9-
32             cout << " "; // 11-13-15-18-19
33         }
34         for (int j = 1; j <= 21 - 2 * i; j++) { //打印下半部分 "*" 19~1 公差为2
35             if (j % 2 == 0) { //偶数挖空
36                 cout << " ";
37             }
```

```

38         else {
39             cout << "*";
40         }
41     }
42     cout << endl;
43 }
44 return 0;
45 }

```



3 a 求两个有序数组的公共元素

```

1  #include<iostream>           //在二分查找和双指针法之间反复横跳 想来想去还是双指针吧
2  #include<vector>             //在两个数组的起始位置设置两个指针
3  using namespace std;        //比较二者大小 谁大谁不懂, 谁小谁往前走
4  int main(){
5      int input_a, input_b ;
6      vector<int> a;           //动态数组暂存
7      vector<int> b;
8      while (cin >> input_a){//连续读入
9          if (input_a != -1) {
10             a.push_back(input_a);
11         }
12         else {
13             break;
14         }
15     }
16     while (cin >> input_b){
17         if (input_b != -1) {
18             b.push_back(input_b);
19         }
20         else {
21             break;
22         }
23     }
24     int i = 0, j = 0; //设置两个指针
25     while (i < a.size() && (j < b.size())){ //谁大谁不动 谁小谁往前走
26         if (a[i] == b[j]) {           //相等则直接输出
27             cout << a[i] << ' ';

```

```

28     i++;
29     j++;
30     continue;
31 }
32 else if (a[i] < b[j]) {           //b大则a的指针++
33     i++;
34 }
35
36     else {                       //a大则b的指针++
37         j++;
38     }
39 }
40 }

```

3.b求三个有序数组的公共元素

```

1  #include<iostream>           //三指针
2  #include<vector>             //都相等则输出
3  using namespace std;         //若不全相等 则找出最大值
4  int Max(int a, int b, int c); //剩下两个指针向右移
5  int main() {
6      int input_a, input_b, input_c;
7      vector<int> a;
8      vector<int> b;
9      vector<int> c;
10     while (cin >> input_a) {
11         a.push_back(input_a);
12     }
13     while (cin >> input_b) {
14         b.push_back(input_b);
15     }
16     while (cin >> input_c) {
17         c.push_back(input_c);
18     }
19     int i = 0, j = 0, k = 0;
20     while (i < a.size() && j < b.size() && k < c.size()) { //谁大谁不动 谁小谁往
        前走
21         if (a[i] == b[j] && a[i] == c[k]) { //相等则直接输出
22             cout << a[i] << ' ';
23             i++;
24             j++;
25             k++;
26             continue;
27         }
28         else if (i == a.size() && j == b.size() && k == c.size()) { //如果某个
        数组取到端点
29             break; //则直接跳
        出循环
30         }
31         else { //剩下的所有情况 都先取最大值 然后两个较小
        的指针移动
32             Max(a[i], b[j], c[k]);
33             if (a[i] == Max(a[i], b[j], c[k])) { //a数组为最大时
34                 j++; //数组b和c的指针移动

```

```

35         k++;
36     }
37     else if (b[j] == Max(a[i], b[j], c[k])) { //b数组为最大时
38         i++; //数组a和c的指针移动
39         k++;
40     }
41     else if (c[k] == Max(a[i], b[j], c[k])) { //c数组为最大时
42         i++; //数组a和b的指针移动
43         j++;
44     }
45 }
46 }
47 return 0;
48 }
49 int Max(int a, int b, int c) {
50     if (a > b && a > c) {
51         return a;
52     }
53     else if (b > a && b > c) {
54         return b;
55     }
56     else {
57         return c;
58     }
59 }

```

3.c求n个有序数组的公共元素

想不出解决n个的办法 归纳不出封装函数解决问题

4.求数组的最大值和次大值

```

1  #include<iostream>
2  #include<vector>
3  using namespace std;
4  int main() {
5      vector<int> a;
6      int input_a;
7      while (cin >> input_a) {
8          a.push_back(input_a);
9      }
10
11     int i = 0;
12     int Max = 0;
13     int Sub_Max = 0;
14     while (i < a.size()){
15         if (Max < a[i] && Sub_Max < a[i]) { //如果都小 重置Max的值并把Max给
Sub_Max
16             Max = a[i];
17             Sub_Max = Max;
18         }
19         if (Max > a[i] && Sub_Max < a[i]) { //如果比Max小但比Sub_Max大 则仅重置
Sub_Max
20             Sub_Max = a[i];
21         }
22         else { //如果都大 则继续循环 i++

```

```

23         continue;
24     }
25     i++;
26 }
27 return 0;
28 }

```

5. 给定一个n个整型元素的数组a，其中有一个元素出现次数超过n / 2，求这个元素

```

1  #include<iostream>           //直接用排序做的
2  #include<vector>             //对数组排序
3  #include<algorithm>          //中间的那个必然是出现频率最高的
4  using namespace std;
5  int main() {
6      int input_a;
7      vector<int> a;           //动态数组暂存
8      while (cin >> input_a) { //连续读入 数组的值
9          if (input_a != -1) {
10             a.push_back(input_a);
11         }
12         else {
13             break;
14         }
15     }
16     int temp;
17     int i = (int)a.size() / 2;
18     for (int i = 0; i < a.size(); i++) { //冒泡排序
19         for (int j = 0; j < a.size() - 1; j++) {
20             if (a[j] > a[j + 1]) { //前值大于后值 则后的值给前
21                 swap(a[j], a[j + 1]);
22             }
23         }
24     }
25     cout << a[i] << endl;
26     return 0;
27 }

```

6. 给定一个含有n个元素的整型数组，找出数组中的两个元素x和y使得abs(x - y)值最小

```

1  #include<iostream> //可能勉强算是贪心算法 大概。。。
2  #include<vector>
3  #include<cmath>
4  using namespace std;
5  int main() {
6      vector<int> a;
7      int input_a;
8      while (cin >> input_a) {
9          a.push_back(input_a);
10     }
11     int x = a[0];
12     int y = a[1];
13     int Min = abs(x - y);

```



```

14     for (int i = 0; i < a.size(); i++) { //遍历数组
15         if (abs(a[i] - a[i + 1]) < Min) { //如果出现比Min更小的 则重置x y Min的
值为当前值
16             x = a[i];
17             y = a[i - 1];
18             Min = abs(a[i] - a[i + 1]);
19         }
20     }
21     cout << "x = " << x << " "
22         << "y = " << y << " "
23         << "abs(x-y) = "
24         << Min << endl;
25     return 0;
26 }

```

7.给定含有1001个元素的数组，其中存放了1-1000之内的整数，只有一个整数是重复的，请找出这个数

```

1  #include <stdio.h>
2  #include <stdlib.h> //仅有一个重复则值为
3  #include <time.h>
4  int main()
5  {
6      int a[1001] = { 0 };
7      for (int i = 0; i < 1000; ++i)
8      {
9          a[i] = i + 1;
10     }
11     srand(time(NULL));
12     for (int i = 1000; i >= 2; --i)
13     {
14         int tmp1 = rand() % i + 1;
15         int tmp2 = a[i - 1];
16         a[i - 1] = a[tmp1 - 1];
17         a[tmp1 - 1] = tmp2;
18     }
19     a[1000] = rand() % 1000 + 1;
20     int Sum = 0;
21     for (int i = 0; i < 1000; i++) {
22         Sum += a[i];
23     }
24     for (int i = 1; i <= 1000; i++) {
25         Sum -= i;
26     }
27     printf("%d\n", &Sum);
28     return 0;
29 }

```