

2.使用sort排序

1、格式

```
1 #include <iostream>
2 #include <string>
3 #include <algorithm>    //头文件是这个
4 using namespace std;
5
6 int main()
7 {
8     int arr[10];
9     for(int i = 0; i < 10; i++)
10    {
11        cin >> arr[i];
12    }
13    sort(arr,arr+10);    //排序, 第一个位置 最后一个位置+1
14    for (int i = 0; i < 10 ;i++)
15    {
16        cout << arr[i] <<" ";
17    }
18    cout <<endl;
19    sort(arr,arr+10,greater<int>());    //从大到小排序
20    for (int i = 0; i < 10 ;i++)
21    {
22        cout << arr[i] <<" ";
23    }
24    cout <<endl;
25    return 0;
26 }
```

2、前K名的平均数（降序排序greater）

- 第一行：为小蒜所在班级的人数 N （其中 $1 \leq N \leq 30$ ）；
- 第二行：为 N 个用 1 个空格隔开的信息学分数（其中分数为 700 及以内正整数）；
- 第三行：老师想计算平均数的尖子生人数 K 。

输出格式

输出一行共一个实数，为信息学分数前 K 名同学的分数的平均数。四舍五入保留两位小数。

样例输入

[复制](#)

```
10
93 85 77 68 59 100 43 94 75 82
4
```

样例输出

[复制](#)

```
93.00
```

```

1  int score[35];    //全局变量初始化的时候为0
2  int main()
3  {
4      int N,K;
5      cin >> N;
6      for (int i = 0; i < N; i++)
7      {
8          cin >> score[i];
9      }
10     cin >> K;
11     sort(score,score+N,greater<int>());
12     int sum = 0;
13     for (int i = 0; i < K;i++)
14     {
15         sum += score[i];
16     }
17     printf("%.2lf", 1.0 * sum / K);
18     return 0;
19 }

```

3、降序排序(排序通用cmp)

```

1  bool cmp(int x, int y)
2  {
3      return x > y;
4  }
5
6  int main()
7  {
8      int arr[10];
9      for (int i = 0; i < 10; i++)
10     {
11         cin >> arr[i];
12     }
13     sort(arr,arr+10,cmp);
14     for (int i = 0; i < 10; i++)
15     {
16         cout << arr[i] << " ";
17     }
18     cout << endl;
19     return 0;
20 }

```

4、整数排序

```

1  bool cmp(int x, int y)
2  {
3      if (x%3 != y%3)
4          return x%3 < y%3;
5      else

```

```
6     return x < y;
7 }
```

5、结构体排序

```
1  #include <iostream>
2  #include <string>
3  #include <algorithm>
4  using namespace std;
5
6  struct Student
7  {
8      string name;
9      int score[4];
10 };
11
12 bool cmp_name(Student x, Student y)
13 {
14     return x.name < y.name;
15 }
16
17 bool cmp_score(Student x, Student y) {
18     if (x.score[0] != y.score[0]){
19         return x.score[0] > y.score[0];
20     }
21     if (x.score[1] != y.score[1]){
22         return x.score[1] > y.score[1];
23     }
24     if (x.score[2] != y.score[2]){
25         return x.score[2] > y.score[2];
26     }
27     return x.score[3] > y.score[3];
28 }
29
30 int main()
31 {
32     Student stu[3];
33     for (int i = 0; i < 3; i++)
34     {
35         cin >> stu[i].name;
36         for (int j = 0; j < 4; j++)
37             cin >> stu[i].score[j];
38     }
39     sort(stu, stu+3, cmp_name);
40     for (int i = 0; i < 3; i++)
41     {
42         cout << stu[i].name << ": ";
43         for (int j = 0; j < 4; j++)
44             cout << stu[i].score[j] << " ";
45         cout << endl;
46     }
```

```

47     sort(stu,stu+3,cmp_score);
48     for (int i = 0; i < 3; i++)
49     {
50         cout << stu[i].name << ": ";
51         for (int j = 0; j <4; j++)
52             cout << stu[i].score[j] << " ";
53         cout <<endl;
54     }
55     return 0;
56 }

```

练习：

6、浮点数排序

```

1  #include <iostream>
2  #include <string>
3  #include <algorithm>
4  #include <cmath>
5  using namespace std;
6
7  const double EPS = 1e-6;
8  double num[105];
9
10 bool cmp(double a,double b)
11 {
12     double da = fabs(a - round(a)); //round是四舍五入的整数
13     double db = fabs(b - round(b));
14     if (fabs(da - db) < EPS)
15         return a < b;
16     else
17         return da < db;
18 }
19
20 int main()
21 {
22     int N;
23     cin >> N;
24     for (int i =0; i <N; i++)
25     {
26         cin >> num[i];
27     }
28     sort(num,num+N,cmp);
29     for (int i =0; i <N; i++)
30     {
31         cout << num[i] << " ";
32     }
33
34     return 0;
35 }

```

7、红绿蓝

```
1  #include <iostream>
2  #include <string>
3  #include <algorithm>
4  #include <cmath>
5  using namespace std;
6  char s[10005];
7  //1红2绿3蓝为一组串成一串串运珠
8  int main()
9  {
10     int len,r,g,b;
11     cin >> s;
12     len = sizeof(s)-1; //sizeof会将空字符\0计算在内,而strlen不会将空字符\0计算
13     sort(s,s+len);
14     cout << s << endl;
15     r = 0;
16     g = 0;
17     b = 0;
18     for(int i = 0; i < len; i++)
19     {
20         if(s[i] == 'R')
21             r++;
22         else if(s[i] == 'G')
23             g++;
24         else
25             b++;
26     }
27     cout << min(min(r,g/2),b/3)<<endl;
28     cout << endl;
29     return 0;
30 }
```

8、整数排序进阶

```
1  bool cmp(int a, int b)
2  {
3      int aa = a, bb = b;
4      int suma = 0, sumb = 0;
5      while(a)
6      {
7          suma += a % 10;
8          a /= 10;
9      }
10     while(b)
11     {
12         sumb += b % 10;
13         b /= 10;
14     }
15     if (suma == sumb)
```

```

16     return aa < bb;
17 else
18     return suma < sumb;
19 }

```

9、摘气球

输入格式

第一行输入两个空格分隔的整数 n, m ($1 \leq n, m \leq 1000$), 其中 n 表示小朋友的数量, m 表示墙上气球的数量。

第二行输入 n 个正整数 (每两个整数之间用空格隔开), 第 i 个数为 a_i ($1 \leq a_i \leq 10^5$), 表示第 i 个小朋友跳起来手能够着的高度为 a_i 。

第三行输入 m 个正整数 (每两个整数之间用空格隔开), 第 i 个数为 h_i ($1 \leq h_i \leq 10^5$), 表示第 i 个气球的高度为 h_i 。

```

1  #include <iostream>
2  #include <string>
3  #include <algorithm>
4  using namespace std;
5  struct Children {
6      int a;
7      int id;
8  };
9  bool cmp(Children a, Children b)
10 {
11     return a.a < b.a;
12 }
13 Children child[1005];
14 int h[1005];    //所有气球的高度
15 int ans[1005];
16 bool used[1005]; //被摘过的气球
17
18 int main()
19 {
20     int n, m;
21     cin >> n >> m;
22     for(int i = 0; i < n; i++)
23     {
24         cin >> child[i].a;
25         child[i].id = i;
26     }
27     for(int i = 0; i < m; i++)
28     {
29         cin >> h[i];
30     }
31     for(int i = 0; i < n; i++)
32     {
33         for (int j = 0; j < m; j++)

```

```
34     {
35         if(!vis[j] && h[j] < child[i].a)
36         {
37             ans[child[i].id]++;
38             vis[j] = true;
39         }
40     }
41 }
42 for (int i =0; i <n; i++) {
43     cout << ans[i] << " ";
44 }
45 return 0;
46 }
```

```
1 return strcmp(a.name,b.name) < 0; //✓
```