结构体

1414. 期末考试成绩排名

期末考试结束了，数学成绩已经出来。

数学老师请你帮忙编写一个程序，可以帮助老师对班级所有同学的考试分数按照由高到低进行排序，并输出按照成绩排序后每个同学的学号、姓名、数学成绩。

输入：

第一行是一个整数 n（ n ≤ 100 ），代表班级的总人数；

接下来 n 行，每行有 3 个数据，第一个数据是某个同学的学号，第二个数据是该同学的姓名的拼音（拼音不含空格），第三个数据是该同学的数学成绩（成绩是整数）；

输出：

按照数学成绩由高到低输出每个同学的学号、姓名、数学成绩，每行含 1 个同学的 3 个数据，3 个数据用空格隔开。（如果出现多个同学数学成绩相同，则按照学号由小到大输出，不存在多个同学学号相同的情况）

输入：

3

1 zhangfang 98

2 liming 100

3 sunhua 99

输出：

2 liming 100

3 sunhua 99

1 zhangfang 98

```cpp
#include <bits/stdc++.h>//1-1414-1 宋老师视频教程
using namespace std;
/* 解法一：使用结构体数组，冒泡排序    按照成绩降序，成绩相同按照学号升序 */
struct Student {
    int num;// 成员变量
    string name;
    int score;
};
int main() {
      Student a[110];
    int i,j,n;
    cin >> n;

    for ( i = 0; i < n; i++) {
        cin >> a[i].num >> a[i].name >> a[i].score;
    }

      for (i = 1; i <= n - 1; i++)          // 冒泡排序
      {
            for (j = 0; j <= n - i - 1; j++)
            {
                  if (a[j].score < a[j + 1].score ||
                        (a[j].score==a[j + 1].score && a[j].num > a[j + 1].num))
                  {
                        swap(a[j], a[j + 1]);
                  }
            }
      }
      // 输出
      for (i = 0; i < n; i++) {
        cout << a[i].num << " " << a[i].name << " " << a[i].score << endl;
      }
    return 0;
}
```

```cpp
#include <bits/stdc++.h>//1-1414-2 宋老师视频教程
using namespace std;
/* 解法二：使用结构体数组，sort 排序！  按照成绩降序，成绩相同按照学号升序 */
struct Student {
    int num;    // 成员变量
    string name;
    int score;
};
bool cmp(Student s1, Student s2) // sort 的比较函数
{
      // 希望满足该规则排序，return true
      if (s1.score > s2.score || (s1.score == s2.score && s1.num < s2.num))
      {
            return true;
      }
      else
      {
            return false;
      }
}

int main() {
      Student a[110];
      int i, j, n;
      cin >> n;

      for (i = 0; i < n; i++) {
            cin >> a[i].num >> a[i].name >> a[i].score;
      }
      sort(a, a + n, cmp);
      for (i = 0; i < n; i++) {
            cout << a[i].num << " " << a[i].name << " "<< a[i].score << endl;
      }
    return 0;
}
```

结构体排序参考解法：

```
#include <bits/stdc++.h>//1-1414-3 推荐学习    javacn
using namespace std;
struct stu{
    int num;
    string name;
    int score;
}a[110];
// 按成绩降序，成绩相同按学号升序
bool cmp(stu s1,stu s2){
      if(s1.score > s2.score || (s1.score == s2.score && s1.num < s2.num))
      {
            return true;
      }
      else{ return false;     }
}
int main()
{
      int n,i;
      cin>>n;
      for(i = 0;i < n;i++)
      {
            cin>>a[i].num>>a[i].name>>a[i].score;
      }
      sort(a,a+n,cmp);

      // 输出
      for(i = 0;i < n;i++)
      {
            cout<<a[i].num<<" "<<a[i].name<<" "<<a[i].score<<endl;
      }
}
```

期末考试成绩排名

冒泡排序参考解法：

```
#include<bits/stdc++.h>//1-1414-4    javacn
using namespace std;
int a[101],c[101];
string b[101];
int main(){
	int n,i,j;
	cin>>n;
	for(i=1;i<=n;i++)
	{
		cin>>a[i]>>b[i]>>c[i];
	}
	// 冒泡排序
	for(i=1;i<=n-1;i++)
	{
		for(j=1;j<=n-i;j++)
		{
			if(c[j] < c[j+1] || c[j]==c[j+1] && a[j]>a[j+1])
			{
				swap(a[j],a[j+1]);
				swap(b[j],b[j+1]);
				swap(c[j],c[j+1]);
			}
		}
	}

	for(i=1;i<=n;i++)
	{
		cout<<a[i]<<" "<<b[i]<<" "<<c[i]<<endl;
	}
	return 0;
}
```

```c
#include<stdio.h>//1-1414-5 475214719
struct student{
	int xh;
	char xm[20];
	int cj;
};
int main()
{
	int i,j,n;
	struct student t;
	scanf("%d",&n);
	struct student stu[n];
	for(i=0;i<n;i++)
	    scanf("%d%s%d",&stu[i].xh,&stu[i].xm,&stu[i].cj);
	for(i=0;i<n;i++)
		for(j=0;j<n-i-1;j++)
			if(stu[j].cj<stu[j+1].cj)
			{
				t=stu[j];
				stu[j]=stu[j+1];
				stu[j+1]=t;
			}

	for(i=0;i<n;i++)
		for(j=0;j<n-i-1;j++)
			if(stu[j].cj==stu[j+1].cj&&stu[j].xh>stu[j+1].xh)
			{
				t=stu[j];
				stu[j]=stu[j+1];
				stu[j+1]=t;
			}

	for(i=0;i<n;i++)
	    printf("%d %s %d\n",stu[i].xh,stu[i].xm,stu[i].cj);
}
```

```c
#include<stdio.h>//1-1414-6   475214719
struct student
{
	int xh;
	char xm[20];
	int cj;
};
int main()
{
	int i,j,n;struct student t;
	scanf("%d",&n);
	struct student stu[n];

	for(i=0;i<n;i++)
	    scanf("%d%s%d",&stu[i].xh,&stu[i].xm,&stu[i].cj);

	for(i=0;i<n;i++)
	    for(j=0;j<n-i-1;j++)
	        if(stu[j].cj<stu[j+1].cj||stu[j].cj==stu[j+1].cj&&stu[j].
xh>stu[j+1].xh)
	        {
	            t=stu[j];
	            stu[j]=stu[j+1];
	            stu[j+1]=t;
	        }
	for(i=0;i<n;i++)
	    printf("%d %s %d\n",stu[i].xh,stu[i].xm,stu[i].cj);
}
```

坐标排序

```
#include<bits/stdc++.h>//2-1490    wupeng
using namespace std;
struct node{
	int x;
	int y;
}a[10010];
int n;

bool cmp(node a, node b){
	if(a.x==b.x)
	{
		return a.y<b.y;
	}
	else
	{
		return a.x<b.x;
	}
}

int main(){
	cin >> n;
	for(int i=1; i<=n; i++)
	{
		cin >> a[i].x >> a[i].y;
	}

	sort(a+1, a+1+n, cmp);

	for(int i=1; i<=n; i++)
	{
		cout << a[i].x << ' '<< a[i].y << endl;
	}
	return 0;
}
```

1315 - 遥控飞机争夺赛

红太阳杯遥控飞机大赛拉开帷幕。比赛规则为，每位选手让自己的飞机从起点到终点飞行 5 次，组委会记录 5 次的飞行的成绩之后去掉一个最大成绩、一个最小成绩后计算剩余 3 个成绩的平值（平均分保留 3 位小数）作为该选手的最终成绩。

有 n 名选手参加了比赛，从键盘读入每位选手的编号以及他们的 5 次飞行的成绩。

请根据 n 名选手的比赛成绩，编程计算出冠军、亚军、季军的编号以及组委会计算出的成绩。（假设不存在多名选手成绩正好一样）（ 4.1.51 ）

输入

第一行为一个整数 n ，代表参加比赛的选手数量（ 4 ≤ n ≤ 100）

后面的 n 行，每行有 6 个数，第一个数是选手的编号，后 5 个数为选手的 5 次飞行的成绩。

输出

3 行：

第一行输出冠军的编号及飞行成绩（保留 3 位小数）用空格隔开 2 个数；

第二行输出亚军的编号及飞行成绩；

第三行输出季军的编号及飞行成绩。

输入

4

11 58 59 60 61 62

18 59 60 61 62 63

23 65 64 63 62 62

10 60 61 61 65 62

输出

23 63.000

10 61.333

18 61.000

```cpp
#include <bits/stdc++.h>//3-1315-1 宋老师视频教程
using namespace std;
struct Player {
    int num;    // 选手编号
    double score;    // 平均成绩
};
bool cmp(Player p1, Player p2) {
      if (p1.score > p2.score){     return true;      }
      else                     {     return false;     }
}
int main() {
      int n, i, j, x, ma, mi, s;
      Player a[110];
      cin >> n;
      for (i = 0; i < n; i++)
      {
            cin >> a[i].num;  // 读入每位选手的编号
            s = 0;// 读入 5 个成绩
            ma = INT_MIN;  // 整数的最小数
            mi = INT_MAX;  // 整数的最大数
            for (j = 0; j < 5; j++)
            {
                  cin >> x;
                  s = s + x;
                  ma = max(ma, x);
                  mi = min(mi, x);
            }
            a[i].score = (s - ma - mi) / 3.0;
      }
      sort(a, a + n, cmp);    // sort(数组的起始地址，数组的结束地址+1)
      for (i = 0; i < 3; i++) {    // 输出前三名
            cout<<a[i].num<<" "<<fixed<<setprecision(3) << a[i].score << endl;
      }
      return 0;
}
```

解法二：结构体数组求解

```
#include<bits/stdc++.h>//3-1315-2  推荐学习    javacn
using namespace std;
struct node{// 思路：用结构体数组存储每个同学的学号、平均分
      int num;// 编号
      double score;// 平均分
};
node a[110];
int n;
int s,ma,mi;// 存储每个人的总分、最高、最低分
int x;// 表示每个人的 5 个分数
bool cmp(node n1,node n2){// 排序
      return n1.score > n2.score;// if(n1.score>n2.score) return true;
} //  else return false;
int main(){
      cin>>n;
      for(int i = 1;i <= n;i++)
      {     cin>>a[i].num;// 编号
            s = 0;// 总分
            ma = INT_MIN;// 最高
            mi = INT_MAX;// 最低
            for(int j = 1;j <= 5;j++)            // 读入 5 个成绩
            {     cin>>x;
                  s = s + x;// 求和
                  ma = max(ma,x);
                  mi = min(mi,x);
            }
            a[i].score = (s - ma - mi) / 3.0;          // 平均分
      }
      sort(a+1,a+n+1,cmp);     // 排序
      for(int i = 1;i <= 3;i++){
            cout<<a[i].num<<" ";     // 输出
            cout<<fixed<<setprecision(3)<<a[i].score<<endl;       }
    return 0;
}
```

遥控飞机争夺赛

```
#include<bits/stdc++.h>//3-1315-3  仅作参考 liuzhong
using namespace std;
struct stu{
	int a;
	float cj[5];
	float pcj;
}s[100];
bool cmp(stu a,stu b) { return a.pcj>b.pcj;  }
int main()
{
	int n;
	cin>>n;
	float maxx,minn;
	for(int i=0;i<n;i++)
	{
		cin>>s[i].a;
		for(int j=0;j<5;j++)  {	cin>>s[i].cj[j];  }
		maxx=0;
		minn=s[i].cj[0];
		for(int o=0;o<5;o++){
			if(s[i].cj[o]>maxx)	{	maxx=s[i].cj[o];  }
			if(s[i].cj[o]<minn)	{	minn=s[i].cj[o];  }
		}
		s[i].pcj=(s[i].cj[0]+s[i].cj[1]+s[i].cj[2]+s[i].cj[3]+s[i].cj[4]-
minn-maxx)/3.0;
	}
	sort(s,s+n,cmp);
	cout<<s[0].a<<" ";
	printf("%.3f\n",s[0].pcj);
	cout<<s[1].a<<" ";
	printf("%.3f\n",s[1].pcj);
	cout<<s[2].a<<" ";
	printf("%.3f\n",s[2].pcj);
}
```

解法一：二维数组求解

```
#include<bits/stdc++.h>//3-1315-4 仅作参考   javacn
using namespace std;
int main(){
	int n,i,j,t,max,min,a[110][7];
	cin>>n;
	for(i=0;i<n;i++){
		// 当前行的总分
		t = 0,max = 1,min = 1;// 此处写的不妥当
		cin>>a[i][0];// 编号
		for(j=1;j<=5;j++){
			cin>>a[i][j];
			t = t + a[i][j];
			if(a[i][j]>a[i][max]){  max = j;    }
			if(a[i][j]<a[i][min]){  min = j;    }
		}
		a[i][6] = t - a[i][max] - a[i][min];
	}
	for(i=1;i<=n;i++){      // 比较成绩  联动编号
		for(j=0;j<n-i;j++){
			if(a[j][6] < a[j+1][6]){
				t = a[j][6];
				a[j][6] = a[j+1][6];
				a[j+1][6] = t;
				t = a[j][0];
				a[j][0] = a[j+1][0];
				a[j+1][0] = t;
			}
		}
	}
	for(i=0;i<3;i++){
		cout<<a[i][0]<<" "<<fixed<<setprecision(3)<<a[i][6]/3.0<<endl;
	}
	return 0;
}
```

分别建两个数组，保存编号和平均值

```
#include <bits/stdc++.h>//3-1315-5    jiangyf70
using namespace std;
int a;
int b[110]; // 存编号
double c[110]; // 存平均值
int main(){
    int n;
    cin >> n;
    for(int i = 0; i < n; i++)
    {
        int maxn = 0, minn = 999, sum = 0;
        cin >> b[i]; // 读入编号
        for(int j = 0; j < 5; j++)
        {
            cin >> a;
            sum += a;
            if(maxn < a) maxn = a;
            if(minn > a) minn = a;
        }
        c[i] = (sum - maxn - minn) / 3.0; // 求平均值
    }
    for(int i = 0; i < n; i++)
    {
        for(int j = 0; j <= n - i; j++)
        {
            if(c[j] < c[j+1]){
                swap(c[j], c[j+1]);  // 冒泡排序
                swap(b[j], b[j+1]);
            }
        }
    }
    for(int i = 0; i < 3;i++){ printf("%d %.3lf\n", b[i], c[i]);}// 输出前三位
      return 0;
}
```

解法二：使用结构体求解

```
#include<bits/stdc++.h>//4-1730-1 推荐学习 javacn
using namespace std;
struct node{
	int money,num;
};
node a[1010];
int m,n;
// 按单价升序排序
bool cmp(node n1,node n2){
	return n1.money<n2.money;
}
int main()
{
	cin>>m>>n;
	for(int i=1;i<=n;i++){
		cin>>a[i].money>>a[i].num;
	}
	sort(a+1,a+n+1,cmp);
	int c=0,s=0;
	for(int i=1;i<=n;i++)
	{
		if(c+a[i].num<m)
		{	// 这家店都买完不够
			c=c+a[i].num;
			s=s+a[i].num*a[i].money;
		 }
		else
		{
			s=s+(m-c)*a[i].money; // 买够 m
			break;
		 }
	 }
	 cout<<s;
}
```

本题显然是使用贪心的思想，既然要花最少的钱购买贺卡，那么必然优先购买价格低的贺卡。将所有店铺的贺卡按价格降序，但要注意数据上要保证价格和数量还是要对应的。

然后从第一家店，依次购买，直到数量满足 m。

购买贺年卡

解法一：使用数组直接排序。

```
#include <bits/stdc++.h>//4-1730-2    javacn
using namespace std;
int m,n;
int price[1100],num[1100],i,j,s,c;
int main() {
	cin>>m>>n;  //m:要买的数量 n:有多少商家
	for(i = 0;i < n;i++){   cin>>price[i]>>num[i];} //读入每家的单价和存货量
	//对单价升序排序，交换单价的同时交换数量
	//排序的次数
	for(i = 1;i <= n - 1;i++){
		for(j = 0;j <= n - i - 1;j++){
			if(price[j] > price[j+1]){
				swap(price[j],price[j+1]);
				swap(num[j],num[j+1]);
			}
		}
	}
	//从第1家开始买
	c = 0; //买了多少
	s = 0;//花了多少钱
	for(i = 0;i < n;i++){
		//先买后退
		c = c + num[i];
		s = s + num[i] * price[i];
		if(c >= m){       //判断要不要退（多买或者正好，循环停止）
			//退掉多买的
			s = s - (c - m) * price[i];
			break;
		}
	}
	cout<<s;
	return 0;
}
```

```c
#include <stdio.h>//4-1730-3    仅作参考  w2ql
void sort(int dj[],int kc[],int n);
int main(){
	int m,n;//m 张贺卡，n 个店铺
	int dj[101],kc[1001];
	scanf("%d%d",&m,&n);
	int i,j;
	for(i=1;i<=n;i++){
		scanf("%d%d",&dj[i],&kc[i]);
	}
	// 按照单价从低到高排序
	sort(dj,kc,n);
	// 尽量从单价最低的购买
	int je=0;// 金额
	for(i=1;i<=n;i++){
		// 购买策略 :kc[i]>=m, 则购买 m 张，结束购买
		//kc[i]<m 则购买 kc[i] 张
		if(kc[i]<m){
			m-=kc[i];// 购买 kc[i] 张
			je+=dj[i]*kc[i];
			kc[i]=0;
		}else{
			kc[i]-=m;// 减库存
			je+=dj[i]*m;// 累加金额
			m=0;// 购买 m 张
			break;
		}
	}
	printf("%d\n",je);
	/*for(i=1;i<=n;i++){
		printf( “%d  “,kc[i]);
	}*/

	return 0;
}
```

```
void sort(int dj[],int kc[],int n){
            int i,j;
            for(i=1;i<=n-1;i++){
                  for(j=1;j<=n-i;j++){
                        if(dj[j]>dj[j+1]){
                              int t=dj[j];
                              dj[j]=dj[j+1];
                              dj[j+1]=t;

                              t=kc[j];
                              kc[j]=kc[j+1];
                              kc[j+1]=t;
                        }
                  }
            }
}
```

1499. 宇宙总统 2

地球历公元 6036 年，全宇宙准备竞选一个最贤能的人当总统，共有 n 个非凡拔尖的人竞选总统，现在投票已经结束，获得选票最多的人将荣登总统的宝座，如果有多个候选人获得票数一致，那么名字字典码最大的人将获得总统的宝座。请你编程计算出谁能够胜任总统的职位。

比如，有 10 个人参加了投票，这 10 张选票结果分别是：

zhangguoqiang，liming，wangfang，zhangguoqiang，wangfang，zhangguoqiangzhaofei，zhaofei，wangfang，zhaofei，

统计结果 zhangguoqiang 、wangfang 、zhaofei 三人每人都是 3 票，由于 zhaofei 名字的字典码最大，zhaofei 当选为本届宇宙总统。

输入：

第 1 行是一个整数 n ，代表投票的总数。（n ≤ 1000）

第 2 行～第 n+1 行，每行是一个得到选票的人的名字（名字一定是小写的拼音，没有空格）。

输出：

输出 n 行，按照每个人的得票数由高到低的顺序输出每个人的名字和得票数，中间用空格隔开。（如果有多个人得票数一致，则名字字典码大的人排在前面）

输入：

10

liming

wangfang

zhangguoqiang

zhangguoqiang

wangfang

zhangguoqiang

zhaofei

zhaofei

wangfang

zhaofei

```cpp
#include <bits/stdc++.h>//5-1499-1 【入门】宇宙总统 2      宋老师视频教程
using namespace std;
/* 思路：使用结构体数组来存储每个人和得票数
         每读入一个名字就在结构体数组中找是否有这个人
         如果有：修改该人的票数 +1
         如果没有：将该人存入结构体数组，默认票数为 1
         最后将结构体数组排序：按得票数降序，得票相同按名字字典码降序 */
struct Node {
      string name;   // 名字
      int count;     // 得票数
} a[1100];
bool cmp(Node n1, Node n2)
{
      if (n1.count > n2.count || (n1.count == n2.count && n1.name > n2.name))
      {
            return true;
      }
      else
      {
            return false;
      }
}
```

```
int main() {
	int i, j, n, k = 0;	// k 代表结构体数组默认的人数
	bool f;
	string s;
	cin >> n;
	for (i = 1; i <= n; i++)	// 读入 n 个人的名字
	{
		cin >> s;  // 读入每个人的名字
		f = false; // 假设找不到
		for (j = 1; j <= k; j++)	// 在结构体数组中查找是否与该人
		{
			if (a[j].name == s) // 存在，更新票数
			{
				a[j].count++;
				f = true;
				break;
			}
		}
		if (f == false) // 如果不存在
		{
			k++;
			a[k].name = s;
			a[k].count = 1;
		}
	}
	sort(a + 1, a + k + 1, cmp);  // 排序
	for (i = 1; i <= k; i++)	{ // 最终 a 数组中有 k 个用户的得票数和名字
	  cout << a[i].name << " " << a[i].count << endl;
	}
	return 0;
}
```

```cpp
#include<bits/stdc++.h>//5-1499-2 结构体数组求解法： 推荐学习 javacn
using namespace std;
struct node {
	string name;
	int num;
} a[1010];
bool cmp(node a,node b) {// 按得票降序，得票相同按名字降序
	if(a.num > b.num || a.num == b.num && a.name > b.name) {return true;}
	else {return false;}
}
int main() {
	int n,k=0;//k 代表结构体数组的下标
	bool f=false;
	cin>>n;
	string s;// 每次得票的人
	for(int i=1; i<=n; i++) {
		f=false;
		cin>>s;
		// 在数组中找一下有没有这个人
		for(int j=1; j<=k; j++) {
			// 找到，增加得票数
			if(a[j].name==s) {
				f=true;
				a[j].num++;
			}
		}
		if(f==false) {                  // 没找到，将该人存入数组
			k++;
			a[k].name=s;
			a[k].num=1;        }
	}
	sort(a+1,a+k+1,cmp);     // 排序
	for(int i=1; i<=k; i++) {      cout<<a[i].name<<" "<<a[i].num<<endl;      }
	return 0;
}
```

宇宙总统 2

```
#include<bits/stdc++.h>//5-1499-3  dragoncatter
using namespace std;
struct node
{
	string s;
	int num;
}a[1010];

int  n;
map<string , int> f;
bool cmp(node a, node b)
{
	if(a.num != b.num) return a.num > b.num;
	else return a.s > b.s;
}
int main()
{
	cin>>n;
	for(int i=1;i<=n;i++)
	{
		cin>>a[i].s;
		f[a[i].s]++;
		a[i].num = f[a[i].s];
	}
	sort(a+1,a+n+1,cmp);
	for(int i= 1;i<=n;i++)
	{
		if(f[a[i].s])
		{
			cout<<a[i].s<<' '<<a[i].num<<endl;
			f[a[i].s] = 0;
		}
	}
}
```

1372. 活动选择

学校在最近几天有 n(n≤100)个活动，这些活动都需要使用学校的大礼堂，在同一时间，礼堂只能被一个活动使。由于有些活动时间上有冲突，学校办公室人员只好让一些活动放弃使用礼堂而使用其他教室。

现在给出 n 个活动使用礼堂的起始时间 $begin_i$ 和结束时间 end_i ($begin_i<end_i$)，请你帮助办公室人员安排一些活动来使用礼堂，要求安排的活动尽量多。请问最多可以安排多少活动？

请注意，开始时间和结束时间均指的是某个小时的 0 分 0 秒，如：3 5 ，指的是 3:00 ~ 5:00，因此 3 5 和 5 9 这两个时间段不算冲突的时间段。

输入

第一行一个整数 n (n ≤ 100)。

接下来的 n 行，每行两个整数，第一个 $begin_i$ ，第二个是 end_i ($begin_i<end_i$ ≤ 32767)。

输出

输出最多能安排的活动数。

输入

11

3 5

1 4

12 14

8 12

0 6

8 11

6 10

5 7

3 8

5 9

2 13

输出

4

/* 思路：定义结构体数组，存储每个活动的起止时间，按照结束时间升序排序

第一个活动必选，遍历后续的所有活动，找到每个开始时间 >= 上一个选中活动结束时间的活动 */

```cpp
#include <bits/stdc++.h>//6-1372-1 【基础】活动选择 宋老师视频教程
using namespace std;
struct Node {// 定义结构体存储每个活动的起止时间 // Node: 节点
	int begin;
	int end;
} a[110];
bool cmp(Node n1, Node n2) // 按照结束时间升序排序
{
	if (n1.end < n2.end) {
		return true;
	}
	else {
		return false;
	}
}
int main() {
	int i, n, c = 0, e;
	cin >> n;
	for (i = 0; i < n; i++) {
		cin >> a[i].begin >> a[i].end;
	}
	sort(a, a + n, cmp);
	c = 1;	// 第一个活动必选
	e = a[0].end;
	for (i = 1; i < n; i++)		// 遍历剩余的活动
	{
		if (a[i].begin >= e)
		{
			c++;
			e = a[i].end;
		}
	}
	cout << c;
	return 0;
}
```

解法二：使用结构体解题

```
#include <bits/stdc++.h>//6-1372-2    推荐学习    javacn
using namespace std;
// 每个时间对
struct node {
	int begin;
	int end;
};
struct node a[110];
int n;
// 按结束时间升序排序
bool cmp(node n1,node n2) {
	return n1.end < n2.end?true:false;
}
int main() {
	int i;
	cin>>n;
	for(i = 0; i < n; i++)
	{
		cin>>a[i].begin>>a[i].end;
	}
	sort(a,a+n,cmp);
	int ans=1;  // 第 1 个活动必选
	int t = a[0].end;
	for(int i=1; i<n; i++)  // 在剩余活动中选择
	{
		// 如果当前活动与之前最后结束的活动不冲突，就接受当前活动。
		if(a[i].begin>=t)
		{
			ans++;
			t=a[i].end;
		}
	}
	cout<<ans;
}
```

贪心的策略：优先选择结束时间尽可能早的活动，因为结束时间越早，剩余的时间就越多，那么剩余的时间可以排更多的活动．解法：按照结束时间升序排序，然后逐个统计哪个活动的开始时间 >= 上一个被选中活动的结束时间　　解法一：使用整数数组排序解题

```
#include <bits/stdc++.h>//6-1372-3  javacn
using namespace std;
int n;// 活动数量
int s[110],e[110];// 活动起止时间
int i,j,c,etime;
int main() {
	cin>>n;
	for(i = 0;i < n;i++)	{	cin>>s[i]>>e[i];	}
	for(i = 1;i <= n - 1;i++)// 按照结束时间升序
	{
		for(j = 0;j <= n - i - 1;j++)
		{
			if(e[j] > e[j+1])
			{
				swap(e[j],e[j+1]);
				swap(s[j],s[j+1]);
			}
		}
	}
	etime = e[0];// 存储第 1 个活动的结束时间	// 第 1 个活动必选
	c = 1;// 已经有 1 个选中的活动了
	for(i = 1;i < n;i++)	// 在剩余活动中继续选
	{
		if(s[i] >= etime)// 如果活动的开始时间 >= 上一个选中活动的结束时间
		{
			c++;
			etime = e[i];
		}
	}
	cout<<c;
	return 0;
}
```

活动选择

```
#include<bits/stdc++.h>//6-1372-4    仅作参考 jiangyf70
using namespace std;
struct meet
{
    int l, r;
}a[105];

bool cmp(meet a, meet b)
{
    if(a.r < b.r) return 1;
    return 0;
}
int main()
{
    int n;
    cin >> n;
    for(int i = 0; i < n; i++) cin >> a[i].l >> a[i].r;
    sort(a, a+n, cmp);
    int cnt = 0;
    for(int i = 1; i < n; i++)
    {
        int j = i;
        while(a[i].r > a[j].l) j++;
        cnt++;
        i = j - 1;

    }
    cout << cnt;
    return 0;
}
```

```cpp
#include<bits/stdc++.h>//7-1740-1  推荐学习 jiangyf70
using namespace std;
struct num
{
    int k, c;
}a[105];
bool cmp(num a, num b)
{
    if(a.k < b.k)return 1;
    return 0;
}
int main()
{
    int n;
    cin >> n;
    int m = 0, x;
    for(int i = 0; i < n; i++)
    {
        cin >> x;
        bool f = 0;
        for(int j = 0; j < m; j++)
        {
            if(a[j].k == x)    {    a[j].c++;    f = 1;    }
        }
        if(f == 0)    {    a[m].k = x;    a[m].c = 1;    m++;    }
    }

    sort(a, a + m, cmp);
    for(int i = 0; i < m; i++)
    {
        cout << a[i].k << " "<< a[i].c << endl;
    }
    return 0;
}
```

```cpp
#include<bits/stdc++.h>//7-1740-2    w2016010182
using namespace std;
int main()
{
	map<int,int,less<int> >prap;
	int n;
	scanf("%d",&n);
	for(int i=1;i<=n;++i)
	{
		int dup;
		scanf("%d",&dup);
		if(prap.count(dup)==0)
		{
			pair<int,int>prap2(dup,1);
			prap.insert(prap2);
		}
		else
		{
			prap[dup]+=1;
		}
	}
	map<int,int>::iterator it;
	for(it=prap.begin();it!=prap.end();it++)
	{
		printf("%d %d\n",it->first,it->second);
	}
	return 0;
}
```

统计每个数出现的次数

思路：将数组排序，统计连续相同的数有几个

1. 遇到每个数都 c++

2. c++ 之后判断连续相同的数是否结束

结束标准：到了最后一个数 || 当前数 != 下一个数

```
#include <bits/stdc++.h>//7-1740-3  javacn
using namespace std;
/*
思路：将数组排序，统计连续相同的数有几个
1. 遇到每个数都 c++
2. c++ 之后判断连续相同的数是否结束
结束标准：到了最后一个数 || 当前数 != 下一个数
*/
int n,a[1010],c = 0;
int main()
{
	cin>>n;
	for(int i = 0;i < n;i++){
		cin>>a[i];
	}
	// 排序
	sort(a,a+n);
	// 统计连续相同的数出现的次数
	for(int i = 0;i < n;i++)
	{
		c++;// 统计连续相同的数出现的次数
		if(i == n - 1 || a[i] != a[i + 1])  // 判断连续相同的数是否结束
		{
			cout<<a[i]<<" "<<c<<endl;	// 输出
			c = 0;		// 计数器清零
		}
	}
	return 0;
}
```

混合牛奶 Mixing Milk 按单价排序结构体，从单价最小的开始采购，直到数量 sum==n 即可

```
#include<bits/stdc++.h>//8-1940    jiangyf70
using namespace std;
struct niu
{
    int p, a;
}a[2000005];
bool cmp(niu a, niu b)
{    return a.p < b.p;}
int main()
{
    int n, m;
    cin >> n >> m;
    for(int i = 0; i < m; i++)  {    cin >> a[i].p >> a[i].a;   }
    sort(a, a + m, cmp);
    int sum = 0, fy = 0, i = 0;
    while(sum < n)
    {
        if(sum + a[i].a <= n)
        {
            fy += a[i].a * a[i].p;
            sum += a[i].a;
        }
        else
        {
            fy = fy + (n - sum) * a[i].p;
            sum = n;
            break;
        }
        i++;
    }
    cout << fy;
    return 0;
}
```

仰卧起坐成绩统计

结构体求解：

```
#include<bits/stdc++.h>//9-1314-1 推荐学习 javacn
using namespace std;
struct node {
	int cnt;
	char grade;
};
node a[10];
int n,x;
bool cmp(node n1,node n2) {
	if(n1.cnt>n2.cnt||(n1.cnt==n2.cnt&&n1.grade<n2.grade)) {	return true;}
	else {return false;	}
}
int main() {
	cin>>n;
	string t="0ABCDEFG";
	for(int i=1; i<=6; i++) {
		a[i].grade=t[i];
		a[i].cnt=0;
	}
	for(int i=1; i<=n; i++)
	{
		cin>>x;
		if(x>=60)a[1].cnt++;
		else if(x>=50)a[2].cnt++;
		else if(x>=40)a[3].cnt++;
		else if(x>=30)a[4].cnt++;
		else if(x>=20)a[5].cnt++;
		else a[6].cnt++;
	}
	sort(a+1,a+6+1,cmp);
	for(int i=1; i<=6; i++) {	cout<<a[i].grade<<":"<<a[i].cnt<<endl;}
	return 0;
}
```

结构体无函数求解

```
#include<bits/stdc++.h>//9-1314-2  推荐学习  475214719
using namespace std;
struct t {
    char ch;
    int cnt;
} s[6] = {'A', 0, 'B', 0, 'C', 0, 'D', 0, 'E', 0, 'F', 0};

int main() {
      int n, a[101];
      cin >> n;
      for(int i = 1; i <= n; i++)
            cin >> a[i];

      for(int i = 1; i <= n; i++)
      {
            if(a[i] >= 60)       s[0].cnt++;
            else if(a[i] >= 50) s[1].cnt++;
            else if(a[i] >= 40) s[2].cnt++;
            else if(a[i] >= 30) s[3].cnt++;
            else if(a[i] >= 20) s[4].cnt++;
            else                s[5].cnt++;
      }

      for(int k = 1; k < 6; k++)
            for(int j = 0; j < 6 - k; j++)
                  if(s[j].cnt < s[j + 1].cnt || s[j].cnt == s[j + 1].cnt &&
s[j].ch > s[j + 1].ch)
                        swap(s[j], s[j + 1]);

      for(int i = 0; i < 6; i++)
            cout << s[i].ch <<":" << s[i].cnt << endl;
    return 0;
}
```

新生舞会

```cpp
#include<bits/stdc++.h>//10-1953-1  推荐学习 jiangyf70
using namespace std;
int n; // 人员数量
struct person
{
    string name;
    string num;
    char c;
}a[1005];
char findp(string s)
{
    for(int i = 0; i < n; i++)
    {
        if(a[i].name == s || a[i].num == s) return a[i].c;
    }
    return '0';
}
int main()
{
    cin >> n;
    for(int i = 0; i < n; i++)  {    cin >> a[i].name >> a[i].num >> a[i].c;   }
    int m;
    cin >> m;
    for(int i = 0; i < m; i++)
    {
        string x, y;
        cin >> x >> y;
        if(findp(x) != findp(y)) cout << "Y" << endl;
        else cout << "N" << endl;
    }
    return 0;
}
```

解题思路：用结构体数组读入 n 个人的信息

m 次询问，每询问一次，就判断一次

定义函数，给定一个字符串，函数中判断字符串是名字还是学号

根据给定的字符串，查找其性别

```
#include <bits/stdc++.h>//10-1953-2    javacn
using namespace std;
// 定义结构体的同时，定义结构体数组
struct stu{
	string name;
	string num;
	char gender;
}a[1010];
int n,m;
string s1,s2;
// 根据给定的信息，查询性别
char fun(string s){
	// 如果是学号
	if(isdigit(s[0]))
	{
		for(int i = 1;i <= n;i++)
		{
			if(a[i].num == s) return a[i].gender;
		}
	}
	else
	{
		// 姓名
		for(int i = 1;i <= n;i++)
		{
			if(a[i].name == s) return a[i].gender;
		}
	}
}
```

```
int main()
{
	cin>>n;
	for(int i = 1;i <= n;i++)
	{
		cin>>a[i].name>>a[i].num>>a[i].gender;
	}

	// 读入 m 对要判断的信息
	cin>>m;
	for(int i = 1;i <= m;i++)
	{
		cin>>s1>>s2;
		if(fun(s1) != fun(s2))
		{
			cout<<"Y"<<endl;
		}
		else
		{
			cout<<"N"<<endl;
		}
	}
	return 0;
}
```