排序练习+结构体

人员

韩承睿、牟茗、刘嘉航、辛帅辰、高健桓、李翰如、崔吉诺、刘祺、夏硕承、秦显森、谢亚锴、王静嘉、牛同泽、徐浩然、刘轩铜 到课， 齐振玮 线上

作业检查

韩承睿 已完成

牟茗 已完成

刘嘉航 已完成

辛帅辰 已完成

高健桓 已完成

李翰如 已完成

方俊喆 未打卡

崔吉诺 已完成

刘祺 已完成

夏硕承 已完成

秦显森 已完成

齐振玮 已完成

谢亚锴 已完成

王静嘉 已完成

牛同泽 已完成

徐浩然 已完成

刘轩铜 已完成

作业

必做: **noi 1.10 06:整数奇偶排序** 和 **noi 1.10 09:明明的随机数** 和 **noi 1.10 10:单词排序**

下节课默写: **东方博宜 1326. 需要安排几位师傅加工零件** 和 **noi 1.10 02:奇数单增序列**

课堂表现

牟茗、高健桓、李翰如、刘祺、辛帅辰、崔吉诺、谢亚锴、齐振玮、刘轩铜、刘嘉航、秦显森、夏硕承、韩承睿, 这些同学独立默写2个题都通过了，提出表扬！！

今天课上讲了结构体内容，结构体在初学时可能会有一点抽象，同学们课下要好好复习。

课堂内容

东方博宜 1453 - 橘子排队

排序，求 a[2] ~ a[n-1] 的平均值，并输出

```
#include <bits/stdc++.h>

using namespace std;

const int maxn = 200 + 5;
int w[maxn];

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

  for (int i = 1; i <= n; ++i) {
    for (int j = i+1; j <= n; ++j) {
      if (w[j] < w[i]) {
        int t = w[i];
        w[i] = w[j];
        w[j] = t;
      }
    }
  }

  int sum = 0;
  for (int i = 2; i <= n-1; ++i) sum += w[i];

  printf("%.1f\n", (double)sum/(n-2));
  for (int i = 2; i <= n-1; ++i) cout << w[i] << " ";
  cout << endl;
  return 0;
}
```

东方博宜 1166 - 数的排序

先将输入的每个数拆位，然后重新排序

```
#include <iostream>

using namespace std;
```

```
int a[15];

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

    for (int i = 1; i <= n; i++) {
        // 拆位，重新改 a[i] 的值
        int sum = 0;
        while (a[i]!=0) {
            int t = a[i]%10;
            a[i]/=10;
            sum += t;
        }

        a[i]=sum;
    }

    for (int i = 1; i <= n; i++) {
        for (int j = i+1; j <= n; j++) {
            if (a[j] < a[i]) {
                int t = a[j];
                a[j] = a[i];
                a[i] = t;
            }
        }
    }

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

东方博宜 1326 - 需要安排几位师傅加工零件

从大到小排序，然后选师傅，直到做的总数超过 m

```
#include <iostream>
using namespace std;

int a[105];

int main() {
    int n, m;
    cin >> n >> m;
```

```
    int sum = 0;
    for (int i = 1; i <= m; i++) {
        cin >> a[i];
        sum += a[i];
    }
    if (sum < n) {
        cout << "NO" << endl;
    }
    else {
        for (int i = 1; i <= m; i++) {
            for (int j = i+1; j <= m; j++) {
                if (a[i] < a[j]) {
                    int t = a[i];
                    a[i] = a[j];
                    a[j] = t;
                }
            }
        }

        int cnt = 0;
        for (int i = 1; i <= m; i++) {
            cnt += a[i];
            if (cnt >= n) {
                cout << i << endl;
                break;
            }
        }
    }
    return 0;
}
```

东方博宜 1497 - 分数线的划定

从大到小排序，输出第 x 个成绩

```
#include <bits/stdc++.h>

using namespace std;

int a[1005];

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

```
        int t = a[i];
        a[i] = a[j];
        a[j] = t;
      }
    }
  }
  cout << a[x] << endl;
  return 0;
}
```

noi 1.10 02:奇数单增序列

```
#include <bits/stdc++.h>

using namespace std;

const int maxn = 500 + 5;
int w[maxn];

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

  for (int i = 1; i <= n; ++i) {
    for (int j = i+1; j <= n; ++j) {
      if (w[i] > w[j]) {
        int t = w[i];
        w[i] = w[j];
        w[j] = t;
      }
    }
  }

  bool flag = false;
  for (int i = 1; i <= n; ++i) {
    if (w[i]%2 == 1) {
      if (flag == false) {
        cout << w[i];
        flag = true;
      } else {
        cout << "," << w[i];
      }
    }
  }
  cout << endl;
  return 0;
}
```

noi 1.10 06:整数奇偶排序

```cpp
// 方法一
#include <iostream>
using namespace std;

int a[105];

int main() {
    int n = 10;
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
    }

    for (int i = 1; i <= n; i++) {
        for (int j = i+1; j <= n; j++) {
            if (a[j] > a[i]) {
                int t = a[j];
                a[j] = a[i];
                a[i] = t;
            }
        }
    }

    for (int i = 1; i <= n; i++) {
        if (a[i]%2 == 1) {
            cout << a[i] << " ";
        }
    }

    for (int i = n; i >= 1; i--) {
        if (a[i]%2 == 0) {
            cout << a[i] << " ";
        }
    }
    return 0;
}
```

```cpp
// 方法二
#include <bits/stdc++.h>

using namespace std;

const int maxn = 200 + 5;
int w[maxn];

int main()
{
  int n = 10;
  for (int i = 1; i <= n; ++i) cin >> w[i];

  for (int i = 1; i <= n; ++i) {
```

```
      for (int j = i+1; j <= n; ++j) {
        if ((w[i]%2==0&&w[j]%2==0&&w[j]<w[i]) ||
            (w[i]%2==1&&w[j]%2==1&&w[j]>w[i]) ||
            (w[i]%2==0&&w[j]%2==1)) {
              int t = w[i];
              w[i] = w[j];
              w[j] = t;
            }
      }
    }

    bool flag = false;
    for (int i = 1; i <= n; ++i) {
      cout << w[i] << " ";
    }
    cout << endl;
    return 0;
}
```

结构体讲解

1. 结构体概念：一种自己新定义的数据类型

struct 结构体名字 { int xxx; double xxx; char xxx; };

例子：

```
struct student {
    int id;
    int score;
    int height;
};

student a[100];
```

上述代码定义了一个 student 类型，代表学生类型(包含id, score, height三个属性)

下面的 a 数组，相当于开了一百个 student 类型的变量

2. 结构体访问

通过 . 的方式进行访问。比如想要访问上述 a 数组的第51个学生的 id 属性，可以采用 a[51].id 的方法访问

3. 结构体输入

cin 时要具体到哪一个属性。比如想要输入上述 a 数组的第51个学生的 id、score、height 三个属性，可以 cin >> a[51].id >> a[51].score >> a[51].height;

4. 结构体交换: 跟 int 的交换方式一致

```
struct student {
    int id, score, height;
}
student a, b;
student t = a;
a = b;
b = t;
```

5. 结构体直接赋值

```
struct student {
    int id, score, height;
}

student a = {15, 89, 165};
```

此时，a.id = 15, a.score = 89, a.height = 165;

结构体总结

```
#include <iostream>
using namespace std;

struct student {
    int yu;
    int shu;
    double ying;
};

int main() {
    student a;
    cin >> a.yu >> a.shu >> a.ying;

    student b = {90, 80, 85};
    cout << b.yu << " " << b.shu << " " << b.ying << endl;

    student t = a;
    a = b;
    b = t;
    return 0;
}
```