

# C语言第六章作业-1

## 公式求和

【问题描述】输入2个正整数a和n，求  $a+aa+aaa+aa\dots a$  ( $n\uparrow a$ ) 之和。例如：输入2和3，输出 246 (2+22+222)。

【输入输出样例】 Input a&n:2 3 s=246 (“=”两边无空格，输出结束不换行)

## 解析

- 其实这里循环相加就好了
  - $aa = a * 10 + a$  根据这个公式我们就可以得到下一项需要累加的值

## 示例代码

```
1 #include <stdio.h>
2 int main() {
3     int a, n, i, sum = 0, term = 0;
4     printf("Input a&n:");
5     scanf("%d %d", &a, &n);
6     for (i = 1; i <= n; i++) {
7         term = term * 10 + a; // 计算下一项
8         sum += term; // 累加
9     }
10    printf("s=%d", sum);
11    return 0;
12 }
```

## 寻找自守数

【问题描述】自守数。自守数是其平方后尾数等于该数自身的自然数。例如：55=25，2525=625。求1~1000之间的所有自守数，每行输出5个数，每个数占6位。

## 解析

- 我们首先需要的遍历 1~1000 之间的所有数，然后再去一个个判断其是否为自守数。
  - 计算这个数的平方的值
  - 判断尾数是否和原数相等（从最后一位依次比较）

## 示例代码

```
1  #include <stdio.h>
2  #include <math.h>
3  int main() {
4      int i, j, n = 0;
5      for (i = 1; i <= 1000; i++) {
6          int square = i * i; //平方
7          int temp = i;
8          int digits = 0;
9          while (temp > 0) { //计算每个数的位数
10             temp /= 10;
11             digits++;
12         }
13         if (square % (int)pow((double)10, (double)digits) == i)
14             { //判断是否是自守数
15                 printf("%6d", i);
16                 n++;
17                 if (n % 5 == 0) { //每5个输出换行
18                     printf("\n");
19                 }
20             }
21         return 0;
22     }
```

## 查找最大值与最小值

【问题描述】利用数组，给定N个整数数据（ $N \leq 100$ ），查找最大值和最小值并输出。

【输入输出样例】

**Input length ( $N \leq 100$ ):** （此处括号、冒号为英文符号，后面无空格；length和左括号间有空格）

6

8 10 13 1 23 7

The max is 23

The min is 1 （此处输出结束不换行，is后面有一个空格）

## 解析

- 首先我们要使用一个长度为100的数组将数据读取进来，n是实际的数组长度
- 同时寻找最大值与最小值
  - 同时将第一个数设置为最大与最小
  - 与后一个数进行比较，若比最大值大则它就是新的最大值，若比最小值小则它就是新的最小值

## 示例代码

```
1  #include <stdio.h>
2  int main() {
3      int N, i, max, min, arr[100];
4      printf("Input length (N<=100):\n");
5      scanf("%d", &N);
6      for (i = 0; i < N; i++)
7          scanf("%d", &arr[i]);
8      max = arr[0];
9      min = arr[0];
10
11     for (i = 1; i < N; i++) {
12         if(arr[i] > max)
13             max = arr[i];
14         if(arr[i] < min)
15             min = arr[i];
16     }
17     printf("The max is %d\n", max);
18     printf("The min is %d", min);
19     return 0;
20 }
```

## 斐波拉契数列的第n项

【例4-5】利用数组，计算并输出Fibonacci数列第i ( $i < 100$ ) 项的数值。

【输入输出样例】 Please select I number to be printed: (此处冒号为英文符号, 后面无空格) 20

The number is 6765 (此处输出结束不换行)

## 解析

- 其实我们只是需要做一个循环计算： $a_n = a_{n-1} + a_{n-2}$ 直至  $n = \text{输入值}$ 
  - 斐波拉契数列的初项一般认定为 $a_0 = 0$ 和 $a_1 = 1$

## 示例代码

```
1 #include <stdio.h>
2 int main() {
3     int i, num;
4     long long int fib[100]; //用long long int代替int用于储存斐波拉
    也数列值，其实int也行
5     printf("Please select I number to be printed:\n");
6     scanf("%d", &i);
7     fib[0] = 0;
8     fib[1] = 1; //斐波那契数列初项
9     for (num = 2; num <= i; num++)
10         fib[num] = fib[num - 1] + fib[num - 2]; //斐波那契数列项
11     printf("The number is %lld", fib[i]); //输出斐波那契数列
12     return 0;
13 }
```

## 计算平均数与标准差

【问题描述】输入20个学生某门课的成绩，要求求这组成绩的平均分和标准差。标准差公

式为：
$$\delta = \sqrt{\frac{1}{N} \sum_{i=0}^N (x_i - \gamma)^2}$$

【输入形式】20个学生的成绩，中间以空格或者回车分开

【输出形式】 **average=%f, std=%f** 输出后不换行

## 解析

- 已经明确的数量情况下如果只是平均数的话我们就不用数组存了，但是这里需要计算标准差，所以依旧需要。
- 开方我们有两种方式都是使用 `math.h` 库里面的函数：
  - 使用 `sqrt()` 函数直接开方。
  - 使用 `pow()` 函数计算原数的 **0.5** 次方。

## 示例代码

```
1 #include <stdio.h>
2 #include <math.h>
3 int main() {
4     double sum = 0, avg, variance = 0, std_deviation,
    scores[21];
5     for (int i = 0; i < 20; i++) {
6         scanf("%lf", &scores[i]); //输入分数
7         sum += scores[i]; //累加
8     }
9     avg = sum / 20; //计算平均值
```

```

10     for (int i = 0; i < 20; i++)
11         variance += pow((double)(scores[i] - avg), (double)2);
12     variance /= 20;
13     std_deviation = sqrt(variance); // 计算标准差
14     printf("average=%f, std=%f ", avg, std_deviation);
15     return 0;
16 }

```

## 数据统计

【问题描述】输入n(n<20)个数，要求在屏幕上输出这n个数中互不相同的那些数，重复的数只输出第一次。（提示：将输入进来的数据中新出现的数写入到另一个数组中）

【输入形式】先输入数字n,再依次输入这n个数

【输出形式】依次输出不相同的数，中间以空格分开。输出结束后不换行。

## 解析

- 我们使用两个数组 `a[n]` 和 `b[n]` 先将数据读入 `a[n]` 再把数据依次放入 `b[n]`
  - `a[n]` 数据放入 `b[n]` 前需要遍历 `b[n]` 看是否已经放入相同数据。
  - 放入完毕后直接输出 `b[n]`

## 示例代码

```

1  #include <stdio.h>
2  int main() {
3      int n, i, j, count = 0, nums[21], unique_nums[21];
4      scanf("%d", &n);
5      for (i = 0; i < n; i++)
6          scanf("%d", &nums[i]); // 读入原始数据
7      unique_nums[count++] = nums[0]; // 将第一个数放入结果数组
8      for (i = 1; i < n; i++) { // 新放入不重复的数，判断
9          int is_duplicate = 0;
10         for (j = 0; j < count; j++) {
11             if (nums[i] == unique_nums[j]) {
12                 is_duplicate = 1; // 当作布尔来使用，0为假，非零为真
13                 break;
14             }
15         }
16         if (!is_duplicate) // 如果不重复就放入结果数组
17             unique_nums[count++] = nums[i];
18     }
19     for (i = 0; i < count; i++) // 输出结果数组
20         printf("%d ", unique_nums[i]);
21     return 0;
22 }

```

# 票数统计

【问题描述】某次选举活动中有5个候选人，其代号分别用 **1~5** 表示；假设有若干选民，每个选民只能选一个候选人，即每张选票上出现的数字只能是1~5间的某一个数字。每张选票上所投候选人的代号由键盘输入，当输入完所有选票后用-1作为终止数据输入的标志。要求统计输出每个候选人的得票数。

【输入形式】输入一串1-5的数字代表各张选票，最后输入-1代表结束

【输出形式】输出几号候选人得到了多少票

【样例输入】 **1 2 1 3 2 4 4 5 1 4 5 5 2 3 3 2 2 4 4 1 5 5 3 2 3 -1**

【样例输出】

**1 candidate:4 tickets**

**2 candidate:6 tickets**

**3 candidate:5 tickets**

**4 candidate:5 tickets**

**5 candidate:5 tickets**

## 解析

- 这个很明显是一个条件循环，我们使用 `while` 循环就好了。
- 这里虽然看起来很像是 `switch` 选择语句的适用范围，但是不要忘记数组是自带序号的，我们只需要用输入的值做索引就好了。

## 示例代码

```
1 #include <stdio.h>
2 int main() {
3     int candidate_votes[5] = {0}; // 5个候选人初始票数为0
4     int vote;
5     while (scanf("%d", &vote) && vote != -1) {
6         if (vote >= 1 && vote <= 5)
7             candidate_votes[vote - 1]++; // 票数+1
8         else
9             printf("Invalid vote. Please enter a number
10             between 1 and 5.\n");
11     }
12     for (int i = 0; i < 5; i++)
13         printf("%d candidate:%d tickets\n", i + 1,
14         candidate_votes[i]); // 输出票数
15     return 0;
16 }
```

# 判断方阵是否对称

【问题描述】输入一个5阶方阵，判断该方阵是否对称（即判断是否所有的 `a[i][j]` 等于 `a[j][i]`）

【输入形式】

【输出形式】 `yes or no`，输出后不换行。

## 解析

- 我们使用一个二维的数组来储存这个方阵，个人还是更建议使用 `char`
  - 使用 `int` 更省心，使用 `char` 要小心读入空格和换行

## 示例代码

```
1 #include <stdio.h>
2 int main() {
3     int matrix[5][5];
4     int is_symmetric = 1;
5     for (int i = 0; i < 5; i++) // 双循环读入2*2的方阵
6         for (int j = 0; j < 5; j++)
7             scanf("%d", &matrix[i][j]);
8     for (int i = 0; i < 5; i++) { // 双循环判断是否为对称
9         for (int j = 0; j < 5; j++) {
10             if (matrix[i][j] != matrix[j][i]) {
11                 is_symmetric = 0;
12                 break;
13             }
14         }
15         if (!is_symmetric)
16             break;
17     }
18     if (is_symmetric)
19         printf("yes");
20     else
21         printf("no");
22     return 0;
23 }
```