

周二下午 16:30 机考题题解

1. 【两数之积的数根】

题目：

- 题目：

对于一个正整数 n ，我们将它的各个位相加得到一个新的数字，如果这个数字是一位数，我们称之为 n 的数根，否则重复处理直到它成为一个一位数，这个一位数也算是 n 的数根

例：考虑 24： $2 + 4 = 6$ ，6 就是 24 的数根

请编写程序，计算两数之积 $n_1 * n_2$ 的数根，输出每一步的计算式

- 输入格式：

一行，一个算式，形如 $n_1 * n_2$

- 输出格式：

若干行：

- 第一行：一个算式，计算 $n_1 * n_2$ 的值，形如 $n_1 * n_2 = ans$
- 之后若干行，每行一个算式，为计算 $n_1 * n_2$ 的数根的一步，形如 $a + b = c$

- 样例输入：

```
11*9
```

- 样例输出：

```
11*9=99
9+9=18
1+8=9
```

思路分析：

1. 先计算 $n_1 * n_2$ ，存在 ans 中，并输出

```
ans = n1 * n2;
printf("%d*d=%d\n", n1, n2, ans);
```

2. 逐步计算 ans 的数根：若 $ans \geq 10$ ，分离 ans 的每一位数，加起来形成新的 ans ，重复直至 $ans < 10$

1. 如何分离 ans 的每一位数？

```
int tmp = ans, cnt = 0, digit[11] = {0};
while (tmp > 0) {
    digit[++cnt] = tmp % 10;
    tmp /= 10;
}
```

此时 `digit[i]` 存储的即是 `ans` 的各位数字

2. 算出“下一个”`ans`:

```
ans = 0;
for (int i=cnt; i>=1; --i) ans += digit[i];
```

3. 如何按照格式输出?

```
for (int i=cnt; i>=1; --i)
    printf("%d", digit[i]);
if (i != 1) printf("+");
else printf("=");
}
printf("%d\n", ans);
```

4. 事实上, 2.2 与 2.3 可以合在一个 `for` 循环中:

```
for (int i=cnt; i>=1; --i) {
    ans += digit[i];
    printf("%d", digit[i]);
    if (i != 1) printf("+");
    else printf("=");
}
printf("%d\n", ans);
```

代码

```
#include <stdio.h>

int main() {
    int n1, n2, ans = 0;
    scanf("%d*d", &n1, &n2);
    ans = n1 * n2;
    printf("%d*d=%d\n", n1, n2, ans);

    while (ans >= 10) {
        int tmp = ans, digit[11] = {0}, cnt = 0;
        ans = 0;
        while (tmp > 0) {
            digit[++cnt] = tmp % 10;
            tmp /= 10;
        }
        for (int i=cnt; i>=1; --i) {
            ans += digit[i];
            printf("%d", digit[i]);
        }
    }
}
```

```
        if (i != 1) printf("+");
        else printf("=");
    }
    printf("%d\n", ans);
}
return 0;
}
```

2. 【欧拉函数】

题目：

- 题目：

欧拉函数 $\varphi(n)$ 表示的是 $1 \sim N$ 之间的数中与 N 互质的数的个数

例： $\varphi(6) = 2$ 是因为 $1 \sim 6$ 中，与 6 互质的数有 1, 5.

请编写程序，输出欧拉函数 $\varphi(N)$ 的值和与 N 互质的 $1 \sim N$ 之间的数

- 输入格式：

一行，一个整数，为题目中的 N

- 输出格式：

共 2 行：

- 第 1 行：一个整数，为 $\varphi(N)$
- 第二行： $\varphi(N)$ 个正整数，用空格隔开，为与 N 互质的 $1 \sim N$ 之间的数

- 样例输入：

6

- 样例输出：

2
1 5

思路分析

1. 如何判断一个数 x 是否与 N 互质？

1. 求 x 与 N 的最大公约数 $\gcd(N, x)$
2. 判断 $\gcd(N, x) == 1$

2. 如何求最大公约数？

1. 背景知识： [辗转相除法](#)

2. 如何写？注意到辗转相除法的特点，可以用 *while* 循环或递归写：

1. 递归写法

```
int gcd(int a, int b) {
    if (a % b == 0) return b;
    else return gcd(b % a, a);
}
```

2. 循环写法

```
int gcd(int a, int b) {
    while (a % b != 0) {
        int t = a % b;
        a = b;
        b = t;
    }
    return b;
}
```

3. 如何计算 $\varphi(N)$? 循环统计与 N 互质的数的个数

```
int cnt = 0;
for (int i=1; i<=N; ++i) {
    if (gcd(i, N) == 1) ++cnt;
}
```

4. 输出

```
printf("%d\n", cnt);
for (int i=1; i<=N; ++i) {
    if (gcd(i, N) == 1) printf("%d ", i);
}
```

代码

```
#include <stdio.h>

int gcd(int a, int b) {
    if (a % b == 0) return b;
    else return gcd(b, a % b);
}

int main() {
    int N;
    scanf("%d", &N);

    int cnt = 0;
    for (int i=1; i<=N; ++i) {
        if (gcd(N, i) == 1) ++cnt;
    }
    printf("%d\n", cnt);
    for(int i=1; i<=N; ++i) {
        if (gcd(N, i) == 1) printf("%d ", i);
    }
}
```

```
    return 0;  
}
```