

# 课时4作业

2025年6月25日 11:19

## 作业一

### Description

输入一个整型数，判断是否是对称数，如果是，输出yes，否则输出no，不用考虑这个整型数过大，int类型存不下，不用考虑负值；  
例如 12321是对称数，输出yes，124421是对称数，输出yes，1231不是对称数，输出no

### Input

一个整型数

### Output

输出是yes，或者no

#### Sample Input 1

12321

#### Sample Output 1

yes

#### Sample Input 2

1231

#### Sample Output 2

no

```
int chapter_four_homework_one() {  
    int input_num = 0;  
    int input_num_reversal = 0;  
  
    scanf("%d", &input_num);  
  
    int input_num_back = input_num;  
  
    // start judge num.  
    while (input_num) {  
        input_num_reversal = input_num_reversal * 10 + input_num % 10;  
        input_num = input_num / 10;  
    }  
  
    input_num_reversal == input_num_back ? printf("yes") : printf("no");  
  
    return 0;  
}
```

## 作业二

### Description

利用while或者for循环计算n! 的值。  
提示:  $n! = 1*2*3*...*n$

### Input

一个正整数n,  $1 \leq n \leq 10$ 。

### Output

n! 的值。

#### Sample Input 1

2

#### Sample Output 1

2

#### Sample Input 2

5

#### Sample Output 2

120

```
int chapter_four_homework_two() {  
    int input_num = 0;  
    int factorial = 1;  
  
    scanf("%d", &input_num);  
}
```

这里如果输入数很大会溢出整型

```

while (input_num) {
    factorial = input_num * factorial;
    input_num--;
}

printf("%d\n", factorial);
return 0;
}

```

### 作业三

#### Description

某人想将手中的一张面值100元的人民币换成10元、5元、2元和1元面值的票子。要求换正好40张，且每种票子至少一张。问：有几种换法？

#### Input

无输入

#### Output

一个数，表示共有多少种换法

#### Sample Input 1

无

#### Sample Output 1

不能告知，因为只有一个数，偷偷告诉你小于10  
0

#### 暴力解法

```

int chapter_four_homework_three() {
    int ten_num = 1;
    int five_num = 1;
    int two_num = 1;
    int one_num = 1;

    int count = 0;
    for (ten_num = 1; ten_num <= 10; ten_num++) {
        for (five_num = 1; five_num <= 20; five_num++) {
            for (two_num = 1; two_num <= 37; two_num++) {
                for (one_num = 1; one_num <= 37; one_num++) {
                    if (ten_num + five_num + two_num + one_num == 40 && ten_num * 10 + five_num * 5 + two_num * 2 + one_num * 1 == 100) {
                        count++;
                    }
                }
            }
        }
    }
    printf("%d\n", count);

    return 0;
}

```

#### 递归解法

```

#include <stdio.h>
int countWays(int total, int numNotes, int notes[], int index, int count) {
    // 如果已经换了足够的票子，并且总金额也匹配
    if (count == 40 && total == 0) {
        return 1;
    }
    // 如果票子数量超过40或总金额不足，返回0
    if (count > 40 || total < 0) {
        return 0;
    }

    int ways = 0;
    for (int i = index; i < numNotes; i++) {
        // 每种票子至少取一张
        ways += countWays(total - notes[i], numNotes, notes, i, count + 1);
    }
    return ways;
}

int main() {
    int notes[] = {10, 5, 2, 1}; // 票子的面值
    int total = 100; // 总金额
    int numNotes = sizeof(notes) / sizeof(notes[0]);

    int result = countWays(total, numNotes, notes, 0, 0);
    printf("换法总数: %d\n", result);

    return 0;
}

```