

C++方向编程题答案

答案说明：

大家如果对本次题目或者答案有问题，可以联系下方的出题老师答疑。

出题老师：

选择题：鲍松山 qq: 365690203

代码题：鲍松山 qq: 365690203

第三周

day18

题目ID：36861-统计每个月兔子的总数

链接：<https://www.nowcoder.com/practice/1221ec77125d4370833fd3ad5ba72395?tpId=37&tqId=21260&rp=1&ru=/activity/oj&ru=/ta/huawei/question-ranking>

【题目解析】：

本题题意明确

【解题思路】：

第n个月的兔子数量由两部分组成，一部分是上个月的兔子 $f(n-1)$ ，另一部是满足3个月大的兔子，会生一只兔子 $f(n-2)$ 。所以第n个月兔子总数： $f(n) = f(n-1) + f(n-2)$ 。本题是在变相考察斐波那契数列。

【示例代码】

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

/*
//迭代法求解
int Fib(int n)
{
    if(n <= 2)
        return 1;
    int f, f1=1, f2=1;
    for(int i=3; i<=n; ++i)
    {
        f = f1 + f2;
        f1 = f2;
        f2 = f;
    }
    return f;
}
*/
```

```
//递归法求解
int Fib(int n)
{
    if(n <= 2)
        return 1;
    else
        return Fib(n-1) + Fib(n-2);
}

int main()
{
    int month;
    int sum = 0;
    while(cin >> month)
    {
        sum = Fib(month);
        cout<<sum<<endl;
    }
    return 0;
}
```

题目ID:36895-字符串通配符

链接: <https://www.nowcoder.com/practice/43072d50a6eb44d2a6c816a283b02036?tpId=37&&tqId=21294&rp=1&ru=/activity/oj&qu=/ta/huawei/question-ranking>

【题目解析】：

本题题意明确，考察字符串匹配。

【解题思路】：

本题可以通过递归求解。从前向后一次匹配，遇到相同字符，都向后移动一个字符，如果通配符遇到"?", 则不需匹配，自动跳过一个字符，如果通配符遇到"*", 则可以匹配任意多个字符，包括0个，此时可以有三种选择，1，匹配0个，通配符向后移动一个字符，字符串不动。2，匹配1个，通配符和字符串都向后移动一个字符。3，匹配多个，通配符不动，字符串向后移动一个字符。

递归的终止条件：通配符或者字符串遇到'\0'。当他们同时结束。

【示例代码】

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

bool match(const char* pattern, const char *str)
{
    //当前字符结束，返回true
    if (*pattern == '\0' && *str == '\0')
        return true;
    //两个字符串有一个先结束，则返回false
    if (*pattern == '\0' || *str == '\0')
        return false;
```

```

if (*pattern == '?')
{
    //新增用例，题目强调?只能够匹配数字和字母，所以增加新的判断
    if(!isdigit(*str) && !isalpha(*str))
        return false;
    //遇到?号，匹配一个字符，跳过一个位置
    return match(pattern + 1, str + 1);
}
else if (*pattern == '*')
{
    // 遇到*号，匹配0个(str不挪动)，1个（两者都向前挪动一个字符）或多个(str向前挪动一个字符)
    while(*pattern=='*')
    {
        pattern++;
    }
    pattern--;
    if(!isdigit(*str) && !isalpha(*str))
        return match(pattern+1,str);

    return match(pattern + 1, str) || match(pattern + 1, str + 1) || match(pattern, str
+ 1);
}
else if (tolower(*pattern) == tolower(*str)) //忽略大小写
{
    //如果当前字符相等，匹配下一个字符
    return match(pattern + 1, str + 1);
}

return false;
}
int main()
{
    string pattern, str;
    while (cin >> pattern >> str)
    {
        bool ret = match(pattern.c_str(), str.c_str());
        if (ret)
            cout << "true" << endl;
        else
            cout << "false" << endl;
    }
    return 0;
}

```