

# 上机题

- 使用下面的方法头编写两个方法：

```
public static int reverse(int number);
```

```
//return the reversal of an integer, i.e., reverse(456) returns 654
```

```
public static Boolean isPalindrome(int number);
```

```
//return true if number is a palindrome (回文数)
```

- 如果一个数字的反向倒置数和它的顺序向数一样，这个数就称为回文数。  
编写一个测试程序，提示用户输入一个整数，然后报告这个整数是否为回文数

# 上机题

- 编写两个方法分别计算下列级数：

- $$m(i) = \frac{1}{2} + \frac{2}{3} + \dots + \frac{i}{i+1}$$

- $$m(i) = 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} \dots + \frac{(-1)^{i+1}}{2i-1}\right)$$

- 编写测试程序，显示两个表格。

i	m(i)
1	0.5000
2	1.1667
3	1.9167
4	2.7167
5	3.5500
6	4.4071
7	5.2821
8	6.1710
9	7.0710
10	7.9801
11	8.8968
12	9.8199
13	10.7484
14	11.6818
15	12.6193
16	13.5604
17	14.5049
18	15.4523
19	16.4023
20	17.3546

i	m(i)
1	4.0000
101	3.1515
201	3.1466
301	3.1449
401	3.1441
501	3.1436
601	3.1433
701	3.1430
801	3.1428
901	3.1427

# 上机题

- 回文素数是指一个数同时为素数和回文数。例如：131是一个素数，同时是一个回文数，数字313和757也是。
- 反素数是一个非回文素数，将其反转之后也是一个素数。例如，17是一个素数，71也是一个素数。
- 如果一个素数可以写成 $2^p - 1$ 的形式，其中p是某个正整数，那么这个素数就被称作梅森素数，例如7。
- 设计并编写相应的方法，借助主函数
  - 显示前若干个回文数
  - 显示前若干个反素数
  - 找出 $p \leq 31$ 的所有梅森素数

要求： 若干的值由用户指定，输出时每10个一行。

# 上机题

- 掷双骰子游戏是赌场中非常流行的游戏。编写程序玩这个游戏的一个变种，具体：
  - 掷两个骰子，每个骰子有六面，分别表示1, 2, …, 6。检查这两个骰子的和。如果和为2、3或12，你就输了；如果和是7或者11，你就赢了；但如果和是其他数，就确定了一个点，继续掷骰子，直到掷出一个7或者掷出和刚才相同的点数。如果掷出的是7，你就输了，如果掷出的点数和你前一次掷出的点数相同，你就赢了。

——设计并实现相关方法

# 上机题

- 编程显示如何测试一个点是否在一个有向直线的左侧、右侧，或是在该直线上。使用下面的方法头编写该方法：

- `public static Boolean leftOfTheLine(double x0,double y0,double x1,double y1,double x2,double y2);`

//point(x2,y2)在由两点(x0,y0)和(x1,y1)确定的直线的左侧，则返回true，否则false

- `public static Boolean onTheSameLine(double x0,double y0,double x1,double y1,double x2,double y2);`

//point(x2,y2)在由两点(x0,y0)和(x1,y1)确定的直线上，则返回true，否则false

- `public static Boolean onTheLineSegment(double x0,double y0,double x1,double y1,double x2,double y2);`

//point(x2,y2)在由两点(x0,y0)和(x1,y1)确定的直线段上，则返回true，否则false

编程提示用户输入三个点p0， p1和p2， 显示p2是否在从p0到p1的直线左侧、右侧、直线上， 或线段上。