

Java第四天练习题

求指定两个数的最大公约数和最小公倍数

答案

```
1  import java.util.Scanner;
2
3  public class Lianxi0804{
4      public static void main(String[] args){
5          Scanner input = new Scanner(System.in);
6          System.out.println("请输入一个数字");
7          int num1 = input.nextInt();
8          System.out.println("请输入另一个数字");
9          int num2 = input.nextInt();
10         int yue = 0; // 定义最大公约数
11         int bei; // 定义最小公倍数
12         for (int i = 1; i ≤ num2; i++) {
13             if (num2 % i == 0 && num1 % i == 0) { // 判断是否可以被
// 两个数整除
14                 yue = i;
15             }
16         }
17         bei = ( num1 * num2 ) / yue;
18         System.out.println("最大公约数为: " + yue);
19         System.out.println("最小公倍数: " + bei);
20     }
21 }
```

打印100以内除了尾数为3,5,7的所有数

答案

```

1 public class Lianxi0804_2 {
2     public static void main(String[] args) {
3         // 打印100以内除了尾数为3,5,7的所有数
4         boolean s = true;
5         for (int i = 1; i ≤ 100; i++){
6             if (i % 10 ≠ 3 && i % 10 ≠ 5 && i % 10 ≠ 7){ // 输出
              不满足这些的数
7                 System.out.println(i);
8             }
9         }
10    }
11 }

```

接收用户输入的数字,判断在此范围内质数的个数, 例如输入90, 输出0-90以内的所有质数的数量

答案

```

1 import java.util.Scanner;
2
3 public class Lianxi0804_3 {
4     public static void main(String[] args) {
5         // 接收用户输入的数字,判断在此范围内质数的个数, 例如输入90, 输出0-90
              以内的所有质数的数量
6         Scanner input = new Scanner(System.in);
7         System.out.println("请输入求质数个数的范围");
8         int num = input.nextInt();
9         int g = 0;
10        outer : for (int i = 1; i ≤ num; i++) {
11            int count = 0; // 统计能整除的数量
12            for (int j = 1; j ≤ i; j++) {
13                // 从1到i的判断整除的组数
14                if (i % j == 0) {
15                    count++;
16                }
17                // 大于2组跳出本次循环
18                if (count > 2) {
19                    continue outer;
20                }
21            }
22            if (count == 2) {
23                g += 1;
24            }

```

```

25     }
26     System.out.println(num + "以内质数为: " + g + "个");
27 }
28 }

```

洪乞丐干10天，收入是多少？

天朝有一个乞丐姓洪，去天桥要钱

第一天要了1块钱

第二天要了2块钱

第三天要了4块钱

第四天要了8块钱

以此类推

```

1  import java.util.Scanner;
2
3  public class Lianxi0804_4 {
4      public static void main(String[] args) {
5          //      洪乞丐干10天，收入是多少？
6          //      天朝有一个乞丐姓洪，去天桥要钱
7          //      第一天要了1块钱
8          //      第二天要了2块钱
9          //      第三天要了4块钱
10         //      第四天要了8块钱
11         //
12         Scanner input = new Scanner(System.in);
13         System.out.println("请输入天数");
14         int day = input.nextInt();
15         int money = 1; // 第一天的总钱
16         int n = 1; // 当天赚的钱
17         // 开始循环
18         for (int i = 1; i < day; i++){
19             n *= 2; // 当天赚的钱
20             money += n; // 总钱数
21         }
22         System.out.println("洪乞丐 " + day + " 天,收入了 " + money +
23         "元");
24     }
25 }

```

李四跑步

李四每天跑步

第一周周一跑 100 米，周二到周天每天比前一天多 100 米。

往后每周以比前一周周一多 100 米。

请问 李四 n （控制台输入）天后跑了多少米

答案

```

1  import java.util.Scanner;
2
3  public class Lianxi0804_5 {
4      public static void main(String[] args) {
5          //      李四跑步
6          //      李四每天跑步,第一周周一跑 100 米, 周二到周天每天比前一天多 100
          米。
7          //      往后每周一比前一周周一多 100 米。请问 李四  $n$ （控制台输入） 天后跑
          了多少米
8          Scanner input = new Scanner(System.in);
9          System.out.println("请输入跑步天数");
10         int n = input.nextInt();
11         // 1. 处理整周
12         // 2. 处理单个天数  $n \% 7$ 
13         int week = n / 7;    // 完整周
14         int sum = 0;    // 总距离
15         int monday = 0;    // 周一的距离
16         for (int i = 0; i < week; i++) {
17             monday = monday + 100; // 每周周一的距离是上周周一的距离 +
          100
18             int day = monday;
19             for (int j = 1; j ≤ 7; j++) { // 计算一周的总距离
20                 sum += day;
21                 day += 100;
22             }
23         }
24         monday = monday + 100;
25         int day = monday;
26         for (int i = 0; i < n % 7; i++) { // 计算  $n \% 7$  天的总距离
27             sum += day;
28             day += 100;
29         }
30         System.out.println( n + " 天, 总共跑了 " + sum + "米");
31     }

```

计算 n 的阶乘

$5! = 5 * 4 * 3 * 2 * 1$ 输入一个数，计算阶乘

答案

```

1  import java.util.Scanner;
2
3  public class Lianxi0804_6 {
4      public static void main(String[] args) {
5          //      计算 n 的阶乘
6          //      5! = 5 * 4 * 3 * 2 * 1 输入一个数，计算阶乘
7          Scanner input = new Scanner(System.in);
8          System.out.println("请输入一个数");
9          int n = input.nextInt();
10         int num = 1;
11         for (int i = 1; i < n + 1; i++){
12             num *= i;
13         }
14         System.out.println(n + " 的阶乘是 " + num);
15     }
16 }
```

输入一个数字，和位数，判断这个数是否回文数字1221 就是回文数字

答案

```

1  public static void main(String[] args) {
2      //      输入一个数字和位数，判断这个数是否回文数字 1221 就是回文数字
3      Scanner scanner = new Scanner(System.in);
4      System.out.print("请输入一个数字: ");
5      int number = scanner.nextInt();
6      System.out.print("请输入数字的位数: ");
7      int digits = scanner.nextInt();
8      // 方法一
9      //      判断是否是回文数字
10     boolean isPalindrome = true;
11     String str = String.valueOf(number);
```

```

12     int length = str.length();
13     if (length != digits) {
14         isPalindrome = false;
15     }
16     for (int i = 0; i < length / 2; i++) {
17         if (str.charAt(i) != str.charAt(length - i - 1)) {
18             isPalindrome = false;
19         }
20     }
21     if (isPalindrome) {
22         System.out.println(number + " 是回文数字");
23     } else {
24         System.out.println(number + " 不是回文数字");
25     }
26 }

```

输入行数 显示如图所示的空心三角形

```

1      *
2     *  *
3    *    *
4   * * * * *

```

答案

```

1  public class Lianxi0804_8 {
2      public static void main(String[] args) {
3          //      输入行数 显示如图所示的空心三角形
4              //      *
5              //     *  *
6              //    *    *
7              //   * * * * *
8              int num = 4;
9              for (int i = 1; i ≤ num; i++) { // 循环行数
10                 for (int j = 1; j ≤ num - i; j++) { // 循环输出空格
11                     System.out.print(" ");
12                 }
13                 if (i > 1 && i < num) {
14                     for (int j = 1; j ≤ i * 2 - 1; j++) { // 循环输出
15                         *
16                             if (j == 1 || j == 2 * i - 1) {
17                                 System.out.print("* ");
18                             } else {

```

```

18         System.out.print(" ");
19     }
20 }
21 } else {
22     for (int j = 1; j ≤ i * 2 - 1; j++) {
23         System.out.print("* ");
24     }
25 }
26 System.out.println();
27 }
28 }
29 }

```

一个循环，打印输出一个由字符构成的正方形图案，其中边长由用户输入确定。要求字符按照以下顺序循环使用：'#', '@', '*'。例如，当边长为 5 时，输出如下：

```

1  # @ @ @ #
2  @ * * * @
3  @ * * * @
4  @ * * * @
5  # @ @ @ #

```

答案

```

1  public static void main(String[] args) {
2      // 一个循环，打印输出一个由字符构成的正方形图案，其中边长由用户输入确定。要求字符按照以下顺序循环使用：**'#' , '@' , '\*'**.
3      int n = 6;
4      System.out.println(Math.pow(12, 2));
5      for (int i = 1; i ≤ n; i++) {
6          if (i == 1 || i == n) {
7              for (int j = 1; j ≤ n; j++) {
8                  if (j == 1 || j == n) {
9                      System.out.print("\t#");
10                 } else {
11                     System.out.print("\t@");
12                 }
13             }
14         } else {
15             for (int j = 1; j ≤ n; j++) {
16                 if (j == 1 || j == n) {

```

```

17         System.out.print("\t@");
18     } else {
19         System.out.print("\t*");
20     }
21 }
22 }
23 System.out.println();
24 }
25 }

```

使用数组实现斐波拉切数列，输入 **n** 输出前 **n** 个数 存储到数组 再输出

答案

```

1  public static void main(String[] args) {
2      // 使用数组实现斐波拉切数列
3      // 输入 n 输出前 n 个数 存储到数组 再输出
4      Scanner sc = new Scanner(System.in);
5      System.out.print("请输入一个数字: ");
6      int number = sc.nextInt();
7      int[] feibo = new int[number];
8      // int prev = 1; // 第一个值
9      // int next = 1; // 第二个值
10     // feibo[0] = prev;
11     // feibo[1] = next;
12     if (number == 1) {
13         feibo[0] = 1;
14     } else if (number == 2) {
15         feibo[0] = 1;
16         feibo[1] = 1;
17     } else {
18         // feibo : 1 1 2 3
19         feibo[0] = 1;
20         feibo[1] = 1; // 1 1 x
21         for (int i = 3; i ≤ number; i++) { // i = 3 i = 4
22             feibo[i - 1] = feibo[i - 3] + feibo[i - 2];
23         }
24         // int curr = prev + next;
25         // feibo[i - 1] = curr;
26         // prev = next;
27         // next = curr;
28     }
29 }

```



```
29         for (int i = 0; i < feibo.length; i++) {  
30             System.out.print(feibo[i] + " ");  
31         }  
32     }
```