

Home

Java

Servlet

JSP

Struts2

Hibernate

Spring

MyBatis

Java WS

C

C++

Python

PHP

Eclipse

VBA

Web

SQL

Misc

Tuy

Java Cơ Bản

Các Khái Niệm Java OOPs

Java String

Bài Tập Java

Danh sách bài tập java

Lời Giải Bài Tập Java Cơ

Bản 1

Giải phương trình bậc 2

Tính giai thừa

Chuyển đổi hệ cơ số

Dãy số Fibonacci

Tìm UCLN và BCNN

Liệt kê tất cả số nguyên tố nhỏ hơn n

Liệt kê n số nguyên tố đầu tiên

Liệt kê tất cả số nguyên tố có 5 chữ số

Phân tích số nguyên

Tính tổng của các chữ số của một số nguyên

Số thuận nghịch

Fibonacci nhỏ hơn n là nguyên tố

Viết chương trình java tính hàm SLOPE trong excel

Lời Giải Bài Tập Java Cơ

Bản 2

Đếm số từ của một câu ký tự

Số lần xuất hiện các từ trong một câu

Sâu s1 chứa sâu s2?

Lời Giải Bài Tập Mảng

(Array) Trong Java

Liệt kê các phần tử xuất hiện trong mảng đúng 1 lần

Liệt kê các phần tử xuất hiện trong mảng đúng 2 lần

Liệt kê số lần xuất hiện của các phần tử trong một mảng

Sắp xếp mảng theo thứ tự tăng dần

Sắp xếp mảng theo thứ tự giảm dần

Chèn phần tử vào mảng trong java

Trộn 2 mảng trong java

Lời Giải Bài Tập Java

Nâng Cao

Lời giải bài tập QLSV

Xử Lý Ngoại Lệ

Các Lớp Lồng Nhau

Đa Luồng (Multithreading)

Java I/O

Bài tập java có lời giải

🔒 Học java core

Lập trình hướng đối tượng(OOPs) trong java 🔒

Bài tập Java có lời giải

Bài này cung cấp cho bạn danh sách các dạng bài tập khác nhau để bạn thực hành khi học java.

1. Bài tập java cơ bản.

2. Bài tập java nâng cao.

```
graph TD; start([start]) --> A[ ]; A --> B[ ]; B --> C[ ]; C --> End([End!]); C --> A;
```

Note: Trước khi xem lời giải thì các bạn hãy tự làm trước nhé và hãy common hóa (phân tách thành phương thức riêng) những gì có thể sử dụng lại được.

1. Bài tập java cơ bản

Trong phần này, bạn phải nắm được các kiến thức về:

- Các mệnh đề if-else, switch-case.
- Các vòng lặp for, while, do-while.
- Các từ khóa break và continue trong java.
- Các toán tử trong java.
- Mảng (array) trong java.
- File I/O trong java.
- Xử lý ngoại lệ trong java.

Bài 01:

Viết chương trình tìm tất cả các số chia hết cho 7 nhưng không phải bội số của 5, nằm trong đoạn 10 và 200 (tính cả 10 và 200). Các số thu được sẽ được in thành chuỗi trên một dòng, cách nhau bằng dấu phẩy.

Gợi ý:

- Sử dụng vòng lặp for

Code mẫu:

```
1 package vn.viettuts.baitap;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Bai01 {
7     public static void main(String[] args) {
8         List<Integer> list = new ArrayList<Integer>();
```

Recent Upc

Bài tập java c

Bài tập quản

Java - Liệt kê

là số nguyên

Tìm số thuận

Java - Tính t

số nguyên n

Java - Phân t

các số nguyê

Java - Liệt kê

chữ số trong

Liệt kê n số r

Java

Java - Liệt kê

hơn n

Java - Tìm ướ

số chung nh

dương

Dãy số Fibon

For-Each tr

VietTuts on

Học

2.980 l

Thích Trang

3 người bạn thíc

https://viettuts.vn/bai-tap-java

1/15

Lập Trình Mạng Với Java

Java Date

Chuyển Đổi Kiểu Dữ Liệu

Java Collections

Java JDBC

Các Tính Năng Mới Trong Java

Câu Hỏi Phỏng Vấn Java

```

9      for (int i = 10; i < 201; i++) {
10         if ((i % 7 == 0) && (i % 5 != 0)) {
11             list.add(i);
12         }
13     }
14     // hiển thị list ra màn hình
15     showList(list);
16 }
17
18 public static void showList(List<Integer> list) {
19     if (list != null && !list.isEmpty()) {
20         int size = list.size();
21         for (int i = 0; i < size - 1; i++) {
22             System.out.print(list.get(i) + ", ");
23         }
24         System.out.println(list.get(size - 1));
25     }
26 }
27 }

```

Kết quả:

```

14, 21, 28, 42, 49, 56, 63, 77, 84, 91, 98, 112, 119, 126, 133, 147, 154, 161, 168,
182, 189, 196

```

Bài 02:

Viết một chương trình tính giai thừa của một số nguyên dương n. Với n được nhập từ bàn phím. Ví dụ, n = 8 thì kết quả đầu ra phải là $1*2*3*4*5*6*7*8 = 40320$.

Gợi ý:

- Sử dụng đệ quy hoặc vòng lặp để tính giai thừa.

Code mẫu: sử dụng đệ quy.

```

1  package vn.viettuts.baitap;
2
3  import java.util.Scanner;
4
5  public class GiaiThuaDemo2 {
6      private static Scanner scanner = new Scanner(System.in);
7      /**
8       * main
9       *
10      * @author viettuts.vn
11      * @param args
12      */
13      public static void main(String[] args) {
14          System.out.print("Nhập số nguyên dương n = ");
15          int n = scanner.nextInt();
16          System.out.println("Giai thừa của " + n + " là: " + tinhGiaiThua(n));
17      }
18      /**
19       * tính giai thừa
20       *
21       * @author viettuts.vn
22       * @param n: số nguyên dương
23       * @return giai thừa của số n
24       */
25      public static long tinhGiaiThua(int n) {
26          if (n > 0) {
27              return n * tinhGiaiThua(n - 1);
28          } else {
29              return 1;
30          }
31      }
32  }
33 }

```

Kết quả:

```

Nhập số nguyên dương n = 8
Giai thừa của 8 là: 40320

```

Bài 03:

Hãy viết chương trình để tạo ra một **map** chứa (i, i^2) , trong đó i là số nguyên từ 1 đến n (bao gồm cả 1 và n), n được nhập từ bàn phím. Sau đó in map này ra màn hình. Ví dụ: Giả sử số n là 8 thì đầu ra sẽ là: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}.

Gợi ý:

- Sử dụng vòng lặp for để lặp i từ 1 đến n .

Code mẫu:

```
1 package vn.viettuts.baitap;
2
3 import java.util.HashMap;
4 import java.util.Map;
5 import java.util.Scanner;
6
7 public class Bai03 {
8     private static Scanner scanner = new Scanner(System.in);
9
10    public static void main(String[] args) {
11        System.out.print("Nhập số nguyên dương n = ");
12        int n = scanner.nextInt();
13
14        Map<Integer, Integer> map = new HashMap<Integer, Integer>();
15        for (int i = 1; i < n + 1; i++) {
16            map.put(i, i * i);
17        }
18        System.out.println(map);
19    }
20 }
```

Kết quả:

```
Nhập số nguyên dương n = 10
{1=1, 2=4, 3=9, 4=16, 5=25, 6=36, 7=49, 8=64, 9=81, 10=100}
```

Bài 04:

Viết chương trình giải phương trình bậc 2: $ax^2 + bx + c = 0$.

Code mẫu:

```
1 package vn.viettuts.baitap;
2
3 import java.util.Scanner;
4
5 /**
6  * Giải phương trình bậc 2
7  *
8  * @author viettuts.vn
9  */
10 public class PhuongTrinhBac2 {
11     private static Scanner scanner = new Scanner(System.in);
12     /**
13      * main
14      *
15      * @param args
16      */
17     public static void main(String[] args) {
18         System.out.print("Nhập hệ số bậc 2, a = ");
19         float a = scanner.nextFloat();
20         System.out.print("Nhập hệ số bậc 1, b = ");
21         float b = scanner.nextFloat();
22         System.out.print("Nhập hằng số tự do, c = ");
23         float c = scanner.nextFloat();
24         giaiPTBac2(a, b, c);
25     }
26
27     /**
28      * Giải phương trình bậc 2:  $ax^2 + bx + c = 0$ 
29      *
30      * @param a: hệ số bậc 2
31      * @param b: hệ số bậc 1
32      * @param c: số hạng tự do
```

```

33      */
34      public static void giaiPTBac2(float a, float b, float c) {
35          // kiểm tra các hệ số
36          if (a == 0) {
37              if (b == 0) {
38                  System.out.println("Phương trình vô nghiệm!");
39              } else {
40                  System.out.println("Phương trình có một nghiệm: "
41                      + "x = " + (-c / b));
42              }
43              return;
44          }
45          // tính delta
46          float delta = b*b - 4*a*c;
47          float x1;
48          float x2;
49          // tính nghiệm
50          if (delta > 0) {
51              x1 = (float) ((-b + Math.sqrt(delta)) / (2*a));
52              x2 = (float) ((-b - Math.sqrt(delta)) / (2*a));
53              System.out.println("Phương trình có 2 nghiệm là: "
54                  + "x1 = " + x1 + " và x2 = " + x2);
55          } else if (delta == 0) {
56              x1 = (-b / (2 * a));
57              System.out.println("Phương trình có nghiệm kép: "
58                  + "x1 = x2 = " + x1);
59          } else {
60              System.out.println("Phương trình vô nghiệm!");
61          }
62      }
63  }

```

Kết quả:

```

Nhập hệ số bậc 2, a = 2
Nhập hệ số bậc 1, b = 1
Nhập hằng số tự do, c = -1
Phương trình có 2 nghiệm là: x1 = 0.5 và x2 = -1.0

```

Bài 05:

Viết chương trình chuyển đổi một số tự nhiên ở hệ số 10 thành một số ở hệ cơ số B ($1 \leq B \leq 32$) bất kỳ. Giả sử hệ cơ số cần chuyển là $2 \leq B \leq 16$. Số đại diện cho hệ cơ số $B > 10$ là A = 10, B = 11, C = 12, D = 13, E = 14, F = 15.

Gợi ý:

- Tham khảo [bảng ASCII](#) để chuyển đổi kiểu char thành String. Hàm `chr(55 + m)` trong ví dụ sau:
- Nếu $m = 10$ trả về chuỗi "A".
- Nếu $m = 11$ trả về chuỗi "B".
- Nếu $m = 12$ trả về chuỗi "C".
- Nếu $m = 13$ trả về chuỗi "D".
- Nếu $m = 14$ trả về chuỗi "E".
- Nếu $m = 15$ trả về chuỗi "F".

Code mẫu:

```

1  package vn.viettuts.baitap;
2
3  import java.util.Scanner;
4
5  public class ConvertNumber {
6      public static final char CHAR_55 = 55;
7      private static Scanner scanner = new Scanner(System.in);
8
9      /**
10       * main
11       *
12       * @author viettuts.vn
13       * @param args
14       */
15      public static void main(String[] args) {
16          System.out.print("Nhập số nguyên dương n = ");
17          int n = scanner.nextInt();
18          System.out.println("Số " + n + " trong hệ cơ số 2 = "
19              + ConvertNumber.convertNumber(n, 2));

```

```

20      System.out.println("Số " + n + " trong hệ cơ số 16 = "
21                          + ConvertNumber.convertNumber(n, 16));
22  }
23
24  /**
25   * chuyển đổi số nguyên n sang hệ cơ số b
26   *
27   * @author viettuts.vn
28   * @param n: số nguyên
29   * @param b: hệ cơ số
30   * @return hệ cơ số b
31   */
32  public static String convertNumber(int n, int b) {
33      if (n < 0 || b < 2 || b > 16) {
34          return "";
35      }
36
37      StringBuilder sb = new StringBuilder();
38      int m;
39      int remainder = n;
40
41      while (remainder > 0) {
42          if (b > 10) {
43              m = remainder % b;
44              if (m >= 10) {
45                  sb.append((char) (CHAR_55 + m));
46              } else {
47                  sb.append(m);
48              }
49          } else {
50              sb.append(remainder % b);
51          }
52          remainder = remainder / b;
53      }
54      return sb.reverse().toString();
55  }
56  }

```

Kết quả:

```

Nhập số nguyên dương n = 15
Số 15 trong hệ cơ số 2 = 1111
Số 15 trong hệ cơ số 16 = F

```

Bài 06:

Dãy số Fibonacci được định nghĩa như sau: $F_0 = 0$, $F_1 = 1$, $F_2 = 1$, $F_n = F_{(n-1)} + F_{(n-2)}$ với $n \geq 2$. Ví dụ: 0, 1, 1, 2, 3, 5, 8, ... Hãy viết chương trình tìm n số Fibonacci đầu tiên.

Code mẫu:

```

1  package vn.viettuts.baitap;
2
3  import java.util.Scanner;
4
5  /**
6   * Tính dãy số Fibonacci bằng phương pháp đệ quy
7   *
8   * @author viettuts.vn
9   */
10 public class FibonacciExample2 {
11     private static Scanner scanner = new Scanner(System.in);
12
13     /**
14      * main
15      *
16      * @param args
17      */
18     public static void main(String[] args) {
19         System.out.print("Nhập số nguyên dương n = ");
20         int n = scanner.nextInt();
21         System.out.println(n + " số đầu tiên của dãy số fibonacci: ");
22         for (int i = 0; i < n; i++) {
23             System.out.print(fibonacci(i) + " ");
24         }
25     }
26
27     /**
28      * Tính số fibonacci thứ n

```

```

29      * @param n: chỉ số của số fibonacci tính từ 0
30      *          vd: F0 = 0, F1 = 1, F2 = 1, F3 = 2
31      * @return số fibonacci thứ n
32      */
33      public static int fibonacci(int n) {
34          if (n < 0) {
35              return -1;
36          } else if (n == 0 || n == 1) {
37              return n;
38          } else {
39              return fibonacci(n - 1) + fibonacci(n - 2);
40          }
41      }
42  }

```

Kết quả:

```

Nhập số nguyên dương n = 12
12 số đầu tiên của dãy số fibonacci:
0 1 1 2 3 5 8 13 21 34 55 89

```

Bài 07:

Viết chương trình tìm ước số chung lớn nhất (USCLN) và bội số chung nhỏ nhất (BSCNN) của hai số nguyên dương a và b nhập từ bàn phím.

Gợi ý:

- Sử dụng [giải thuật Euclid](#).

Code mẫu:

```

1  package vn.viettuts.baitap;
2
3  import java.util.Scanner;
4
5  public class USCLN_BSCNN_1 {
6      private static Scanner scanner = new Scanner(System.in);
7      /**
8       * main
9       *
10      * @param args
11      */
12      public static void main(String[] args) {
13          System.out.print("Nhập số nguyên dương a = ");
14          int a = scanner.nextInt();
15          System.out.print("Nhập số nguyên dương b = ");
16          int b = scanner.nextInt();
17          // tính USCLN của a và b
18          System.out.println("USCLN của " + a + " và " + b
19              + " là: " + USCLN(a, b));
20          // tính BSCNN của a và b
21          System.out.println("BSCNN của " + a + " và " + b
22              + " là: " + BSCNN(a, b));
23      }
24
25      /**
26       * Tìm ước số chung lớn nhất (USCLN)
27       *
28       * @param a: số nguyên dương
29       * @param b: số nguyên dương
30       * @return USCLN của a và b
31       */
32      public static int USCLN(int a, int b) {
33          if (b == 0) return a;
34          return USCLN(b, a % b);
35      }
36
37      /**
38       * Tìm bội số chung nhỏ nhất (BSCNN)
39       *
40       * @param a: số nguyên dương
41       * @param b: số nguyên dương
42       * @return BSCNN của a và b
43       */
44      public static int BSCNN(int a, int b) {
45          return (a * b) / USCLN(a, b);
46      }
47  }

```

Kết quả:

```
Nhập số nguyên dương a = 10
Nhập số nguyên dương b = 24
USCLN của 10 và 24 là: 2
BSCNN của 10 và 24 là: 120
```

Bài 08:

Viết chương trình liệt kê tất cả các số nguyên tố nhỏ hơn n. Số nguyên dương n được nhập từ bàn phím.

Code mẫu:

```
1  package vn.viettuts.baitap;
2
3  import java.util.Scanner;
4
5  /**
6   * Chương trình liệt kê tất cả các số nguyên tố nhỏ hơn n.
7   *
8   * @author viettuts.vn
9   */
10 public class BaiTap08 {
11     private static Scanner scanner = new Scanner(System.in);
12
13     /**
14      * main
15      *
16      * @param args
17      */
18     public static void main(String[] args) {
19         System.out.print("Nhập n = ");
20         int n = scanner.nextInt();
21         System.out.printf("Tất cả các số nguyên tố nhỏ hơn %d là: \n", n);
22         if (n >= 2) {
23             System.out.print(2);
24         }
25         for (int i = 3; i < n; i+=2) {
26             if (isPrimeNumber(i)) {
27                 System.out.print(" " + i);
28             }
29         }
30     }
31
32     /**
33      * check so nguyen to
34      *
35      * @author viettuts.vn
36      * @param n: so nguyen duong
37      * @return true la so nguyen so,
38      *         false khong la so nguyen to
39      */
40     public static boolean isPrimeNumber(int n) {
41         // so nguyen n < 2 khong phai la so nguyen to
42         if (n < 2) {
43             return false;
44         }
45         // check so nguyen to khi n >= 2
46         int squareRoot = (int) Math.sqrt(n);
47         for (int i = 2; i <= squareRoot; i++) {
48             if (n % i == 0) {
49                 return false;
50             }
51         }
52         return true;
53     }
54 }
```

Kết quả:

```
Nhập n = 100
Tất cả các số nguyên tố nhỏ hơn 100 là:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

Bài 09:

Viết chương trình liệt kê n số nguyên tố đầu tiên trong java. Số nguyên dương n được nhập từ bàn phím.

Code mẫu:

```

1  package vn.viettuts.baitap;
2
3  import java.util.Scanner;
4
5  /**
6   * Chương trình liệt kê n số nguyên tố đầu tiên.
7   *
8   * @author viettuts.vn
9   */
10 public class BaiTap09 {
11     private static Scanner scanner = new Scanner(System.in);
12
13     /**
14      * main
15      *
16      * @param args
17      */
18     public static void main(String[] args) {
19         System.out.print("Nhập n = ");
20         int n = scanner.nextInt();
21         System.out.printf("%d số nguyên tố đầu tiên là: \n", n);
22         int dem = 0; // đếm số số nguyên tố
23         int i = 2; // tìm số nguyên tố bắt đầu từ số 2
24         while (dem < n) {
25             if (isPrimeNumber(i)) {
26                 System.out.print(i + " ");
27                 dem++;
28             }
29             i++;
30         }
31     }
32
33     /**
34      * check so nguyen to
35      *
36      * @author viettuts.vn
37      * @param n: so nguyen duong
38      * @return true la so nguyen so,
39      *         false khong la so nguyen to
40      */
41     public static boolean isPrimeNumber(int n) {
42         // so nguyen n < 2 khong phai la so nguyen to
43         if (n < 2) {
44             return false;
45         }
46         // check so nguyen to khi n >= 2
47         int squareRoot = (int) Math.sqrt(n);
48         for (int i = 2; i <= squareRoot; i++) {
49             if (n % i == 0) {
50                 return false;
51             }
52         }
53         return true;
54     }
55 }

```

Kết quả:

```

Nhập n = 10
10 số nguyên tố đầu tiên là:
2 3 5 7 11 13 17 19 23 29

```

Bài 10:

Viết chương trình liệt kê tất cả số nguyên tố có 5 chữ số trong java.

Code mẫu:

```

1  package vn.viettuts.baitap;
2
3  /**
4   * Chương trình liệt kê tất cả số nguyên tố có 5 chữ số.
5   *
6   * @author viettuts.vn

```



```

7  */
8  public class BaiTap10 {
9
10     /**
11     * main
12     *
13     * @param args
14     */
15     public static void main(String[] args) {
16         int count = 0;
17         System.out.println("Liệt kê tất cả số nguyên tố có 5 chữ số:");
18         for (int i = 10001; i < 99999; i+=2) {
19             if (isPrimeNumber(i)) {
20                 System.out.println(i);
21                 count++;
22             }
23         }
24         System.out.println("Tổng các số nguyên tố có 5 chữ số là: " + count);
25     }
26
27     /**
28     * check so nguyen to
29     *
30     * @author viettuts.vn
31     * @param n: so nguyen duong
32     * @return true la so nguyen so,
33     *         false khong la so nguyen to
34     */
35     public static boolean isPrimeNumber(int n) {
36         // so nguyen n < 2 khong phai la so nguyen to
37         if (n < 2) {
38             return false;
39         }
40         // check so nguyen to khi n >= 2
41         int squareRoot = (int) Math.sqrt(n);
42         for (int i = 2; i <= squareRoot; i++) {
43             if (n % i == 0) {
44                 return false;
45             }
46         }
47         return true;
48     }
49 }

```

Kết quả:

```

Liệt kê tất cả số nguyên tố có 5 chữ số:
10007
10009
10037
...
99971
99989
99991
Tổng các số nguyên tố có 5 chữ số là: 8363

```

Bài 11:

Viết chương trình phân tích số nguyên n thành các thừa số nguyên tố trong java. Ví dụ: 100 = 2x2x5x5.

Code mẫu:

```

1  package vn.viettuts.baitap;
2
3  import java.util.ArrayList;
4  import java.util.List;
5  import java.util.Scanner;
6
7  /**
8   * Chương trình phân tích số nguyên n thành các thừa số nguyên tố.
9   * Ví dụ: 12 = 2 x 2 x 3.
10   *
11   * @author viettuts.vn
12   */
13  public class BaiTap11 {
14      private static Scanner scanner = new Scanner(System.in);
15
16      /**
17       * main
18       *

```

```

19     * @param args
20     */
21     public static void main(String[] args) {
22         System.out.print("Nhập số nguyên dương n = ");
23         int n = scanner.nextInt();
24         // phân tích số nguyên dương n
25         List<Integer> listNumbers = phanTichSoNguyen(n);
26         // in kết quả ra màn hình
27         System.out.printf("Kết quả: %d = ", n);
28         int size = listNumbers.size();
29         for (int i = 0; i < size - 1; i++) {
30             System.out.print(listNumbers.get(i) + " x ");
31         }
32         System.out.print(listNumbers.get(size - 1));
33     }
34
35     /**
36     * Phân tích số nguyên thành tích các thừa số nguyên tố
37     *
38     * @param positiveInt
39     * @return
40     */
41     public static List<Integer> phanTichSoNguyen(int n) {
42         int i = 2;
43         List<Integer> listNumbers = new ArrayList<Integer>();
44         // phân tích
45         while (n > 1) {
46             if (n % i == 0) {
47                 n = n / i;
48                 listNumbers.add(i);
49             } else {
50                 i++;
51             }
52         }
53         // nếu listNumbers trống thì add n vào listNumbers
54         if (listNumbers.isEmpty()) {
55             listNumbers.add(n);
56         }
57         return listNumbers;
58     }
59 }

```

Kết quả:

```

Nhập số nguyên dương n = 100
Kết quả: 100 = 2 x 2 x 5 x 5

```

Bài 12:

Viết chương trình tính tổng của các chữ số của một số nguyên n trong java. Số nguyên dương n được nhập từ bàn phím. Với n = 1234, tổng các chữ số: 1 + 2 + 3 + 4 = 10

Code mẫu:

```

1 package vn.viettuts.baitap;
2
3 import java.util.Scanner;
4
5 /**
6  * Chương trình tính tổng của các chữ số của một số nguyên dương n.
7  * Tổng của các chữ số của 6677 là 6 + 6 + 7 + 7 = 26.
8  *
9  * @author viettuts.vn
10 */
11 public class BaiTap12 {
12     private static Scanner scanner = new Scanner(System.in);
13     public static int DEC_10 = 10;
14
15     /**
16     * main
17     *
18     * @param args
19     */
20     public static void main(String[] args) {
21         System.out.print("Nhập số nguyên dương n = ");
22         int n = scanner.nextInt();
23         System.out.printf("Tổng của các chữ số "
24             + "của %d là: %d", n, totalDigitsOfNumber(n));
25     }

```

```

26
27  /**
28   * Tính tổng của các chữ số của một số nguyên dương
29   *
30   * @param n: số nguyên dương
31   * @return
32   */
33  public static int totalDigitsOfNumber(int n) {
34      int total = 0;
35      do {
36          total = total + n % DEC_10;
37          n = n / DEC_10;
38      } while (n > 0);
39      return total;
40  }
41  }

```

Kết quả:

Nhập số nguyên dương n = 6677
 Tổng của các chữ số của 6677 là: 26

Bài 13:

Viết chương trình kiểm tra một số n là số thuận nghịch trong java. Số nguyên dương n được nhập từ bàn phím.

Code mẫu:

```

1  package vn.viettuts.baitap;
2
3  import java.util.Scanner;
4
5  /**
6   * Chương trình liệt kê tất cả các số thuận nghịch có 6 chữ số.
7   *
8   * @author viettuts.vn
9   */
10 public class BaiTap13 {
11     private static Scanner scanner = new Scanner(System.in);
12     /**
13      * main
14      *
15      * @param args
16      */
17     public static void main(String[] args) {
18         System.out.print("Nhập số nguyên dương n = ");
19         int n = scanner.nextInt();
20         System.out.println(n + " là số thuận nghịch: " + isThuanNghich(n));
21         System.out.print("Nhập số nguyên dương m = ");
22         int m = scanner.nextInt();
23         System.out.println(m + " là số thuận nghịch: " + isThuanNghich(m));
24     }
25
26     /**
27      * Kiểm tra số thuận nghịch
28      *
29      * @param n: số nguyên dương
30      * @return true là số thuận nghịch
31      *         false không là số thuận nghịch
32      */
33     public static boolean isThuanNghich(int n) {
34         // chuyển đổi số n thành một chuỗi String
35         String numberStr = String.valueOf(n);
36         // kiểm tra tính thuận nghịch
37         int size = numberStr.length();
38         for (int i = 0; i < (size/2); i++) {
39             if (numberStr.charAt(i) != numberStr.charAt(size - i - 1)) {
40                 return false;
41             }
42         }
43         return true;
44     }
45 }

```

Kết quả:

```
Nhập số nguyên dương n = 123321
123321 là số thuận nghịch: true
Nhập số nguyên dương m = 123451
123321 là số thuận nghịch: false
```

Bài 14:

Viết chương trình liệt kê các số Fibonacci nhỏ hơn n là số nguyên tố trong java. N là số nguyên dương được nhập từ bàn phím.

Code mẫu:

```
1  package vn.viettuts.baitap;
2
3  import java.util.Scanner;
4
5  /**
6   * Chương trình liệt kê các số Fibonacci nhỏ hơn n là số nguyên tố.
7   * Với n được nhập từ bàn phím.
8   *
9   * @author viettuts.vn
10  */
11  public class BaiTap14 {
12      private static Scanner scanner = new Scanner(System.in);
13      /**
14       * main
15       *
16       * @param args
17       */
18      public static void main(String[] args) {
19          System.out.print("Nhập số tự nhiên n = ");
20          int n = scanner.nextInt();
21          System.out.printf("Các số fibonacci nhỏ hơn %d và "
22              + "là số nguyên tố: ", n);
23          int i = 0;
24          while (fibonacci(i) < 100) {
25              int fi = fibonacci(i);
26              if (isPrimeNumber(fi)) {
27                  System.out.print(fi + " ");
28              }
29              i++;
30          }
31      }
32
33      /**
34       * Tính số fibonacci thứ n
35       *
36       * @param n: chỉ số của số fibonacci tính từ 0
37       *          vd: F0 = 0, F1 = 1, F2 = 1, F3 = 2
38       * @return số fibonacci thứ n
39       */
40      public static int fibonacci(int n) {
41          if (n < 0) {
42              return -1;
43          } else if (n == 0 || n == 1) {
44              return n;
45          } else {
46              return fibonacci(n - 1) + fibonacci(n - 2);
47          }
48      }
49
50      /**
51       * check so nguyen to
52       *
53       * @author viettuts.vn
54       * @param n: so nguyen duong
55       * @return true la so nguyen so,
56       *          false khong la so nguyen to
57       */
58      public static boolean isPrimeNumber(int n) {
59          // so nguyen n < 2 khong phai la so nguyen to
60          if (n < 2) {
61              return false;
62          }
63          // check so nguyen to khi n >= 2
64          int squareRoot = (int) Math.sqrt(n);
65          for (int i = 2; i <= squareRoot; i++) {
66              if (n % i == 0) {
67                  return false;
68              }
69          }
70      }
```

```
70     return true;
71 }
72 }
```

Kết quả:

Nhập số tự nhiên n = 100
Các số fibonacci nhỏ hơn 100 và là số nguyên tố: 2 3 5 13 89

Các bài tập khác:

- Viết chương trình nhập số nguyên dương n và thực hiện các chức năng sau:
 - Tính tổng các chữ số của n.
 - Phân tích n thành tích các thừa số nguyên tố.
 - Liệt kê các ước số của n.
 - Liệt kê các ước số là nguyên tố của n.
- Viết chương trình liệt kê các số nguyên có từ 5 đến 7 chữ số thỏa mãn:
 - Là số nguyên tố.
 - Là số thuận nghịch.
 - Mỗi chữ số đều là số nguyên tố.
 - Tổng các chữ số là số nguyên tố.
- Viết chương trình liệt kê các số nguyên có 7 chữ số thỏa mãn:
 - Là số nguyên tố.
 - Là số thuận nghịch.
 - Mỗi chữ số đều là số nguyên tố.
 - Tổng các chữ số là số thuận nghịch.

Bài tập chuỗi trong Java

Danh sách bài tập:

- Nhập một sô ký tự. Đếm số từ của sô đó (mỗi từ cách nhau bởi một khoảng trắng có thể là một hoặc nhiều dấu cách, tab, xuống dòng). Ví dụ " hoc java co ban den nang cao " có 7 từ.
Lời giải: [Đếm số từ trong một chuỗi.](#)
- Nhập một sô ký tự. Liệt kê số lần xuất hiện của các từ của sô đó.
Lời giải: [Liệt kê số lần xuất hiện của các từ trong một chuỗi.](#)
- Nhập 2 sô ký tự s1 và s2. Kiểm tra xem sô s1 có chứa s2 không?
Lời giải: [Chuỗi chứa chuỗi trong java.](#)

Bài tập mảng trong Java

Các bài tập trong phần này thao tác với mảng một chiều và 2 chiều trong java, bạn có thể tham khảo bài học [mảng \(Array\) trong java](#)

Danh sách bài tập:

- Nhập một mảng số nguyên $a_0, a_1, a_2, \dots, a_{n-1}$. In ra màn hình các phần tử xuất hiện trong mảng đúng 1 lần.
Lời giải: [Liệt kê các phần tử xuất hiện trong mảng đúng 1 lần](#)
- Nhập một mảng số nguyên $a_0, a_1, a_2, \dots, a_{n-1}$. In ra màn hình các phần tử xuất hiện trong mảng đúng 2 lần.
Lời giải: [Liệt kê các phần tử xuất hiện trong mảng đúng 2 lần](#)
- Nhập một mảng số nguyên $a_0, a_1, a_2, \dots, a_{n-1}$. In ra màn hình số lần xuất hiện của các phần tử.
Lời giải: [Liệt kê số lần xuất hiện của các phần tử trong một mảng](#)
- Nhập một mảng số nguyên $a_0, a_1, a_2, \dots, a_{n-1}$. Hãy sắp xếp mảng theo thứ tự tăng dần.
Lời giải: [Sắp xếp mảng theo thứ tự tăng dần](#)
- Nhập một mảng số nguyên $a_0, a_1, a_2, \dots, a_{n-1}$. Hãy sắp xếp mảng theo thứ tự giảm dần.
Lời giải: [Sắp xếp mảng theo thứ tự giảm dần](#)

24. Nhập một mảng số nguyên $a_0, a_1, a_2, \dots, a_{n-1}$. Hãy sắp xếp mảng theo thứ tự tăng dần, sau đó chèn phần tử x vào mà vẫn đảm bảo mảng là tăng dần.

Lời giải: [Chèn phần tử vào mảng trong java](#)

25. Nhập một mảng số thực $a_0, a_1, a_2, \dots, a_{n-1}$. Không dùng thêm mảng số thực nào khác (có thể dùng thêm mảng số nguyên), hãy in ra màn hình mảng trên theo thứ tự tăng dần.

26. Nhập 2 mảng số thực $a_0, a_1, a_2, \dots, a_{m-1}$ và $b_0, b_1, b_2, \dots, b_{n-1}$. Giả sử 2 mảng này đã được sắp xếp tăng dần. Hãy tận dụng tính sắp xếp của 2 dãy và tạo dãy $c_0, c_1, c_2, \dots, c_{m+n-1}$ là hợp của 2 dãy trên sao cho c_i cũng có thứ tự tăng dần.

Lời giải: [Trộn 2 mảng trong java](#)

27. Viết chương trình nhập vào mảng A có n phần tử, các phần tử là số nguyên lớn hơn 0 và nhỏ hơn 100.

Thực hiện các chức năng sau:

- Tìm phần tử lớn thứ nhất và lớn thứ 2 trong mảng với các chỉ số của chúng (chỉ số đầu tiên tìm được).
- Sắp xếp mảng theo thứ tự tăng dần.
- Nhập số nguyên x và chèn x vào mảng A sao cho vẫn đảm bảo tính tăng dần cho mảng A.

28. Viết chương trình nhập vào ma trận A có n dòng, m cột, các phần tử là số nguyên lớn hơn 0 và nhỏ hơn 100. Thực hiện các chức năng sau:

- Tìm phần tử lớn thứ nhất với chỉ số của nó (chỉ số đầu tiên tìm được).
- Tìm và in ra các phần tử là số nguyên tố của ma trận (các phần tử không nguyên tố thì thay bằng số 0).
- Sắp xếp tất cả các cột của ma trận theo thứ tự tăng dần và in kết quả ra màn hình.
- Tìm cột trong ma trận có nhiều số nguyên tố nhất.

2. Bài tập java nâng cao

Trong phần này, bạn phải nắm được các kiến thức về:

- Lớp và đối tượng trong java.
- Access modifier trong java
- Các tính chất của lập trình hướng đối tượng (OOP).
- Các khái niệm Java OOPs.
- Collection trong java.
- Xử lý ngoại lệ trong java.

Bài 1: Làm quen với lập trình hướng đối tượng

Đề bài: Viết chương trình quản lý sinh viên. Mỗi đối tượng sinh viên có các thuộc tính sau: id, name, age, address và gpa (điểm trung bình). Yêu cầu: tạo ra một menu với các chức năng sau:

```
/******/
```

- Add student.
- Edit student by id.
- Delete student by id.
- Sort student by gpa.
- Sort student by name.
- Show student.
- Exit.

```
/******/
```

Lời giải: [Bài tập quản lý sinh viên trong java](#)

3 bình luận

Sắp xếp theo

Cũ nhất



Thêm bình luận...

**Nguyễn Bằng**

Sao để bài tập có lời giải mà không thấy lời giải

Thích · Phản hồi · 31 tuần

**Nguyễn Thúy Hiền**

có lời giải mà b

Thích · Phản hồi · 31 tuần

**Nguyễn Văn Quân**

Nguyễn Thúy Hiền lời giải ở đâu vậy ạ ?

Thích · Phản hồi · 22 tuần

**Trương Phúc Khuê**

Bài 1: Làm quen với lập trình hướng đối tượng

rời bài 2 3 4 đâu

Thích · Phản hồi · 27 tuần

**Phuong Nguyen**

bài giải ở phần bên tay trái ấy mọi người

Thích · Phản hồi · 1 · 16 tuần

[Plugin bình luận trên Facebook](#)

Danh sách bài học

[Học java](#)
[Học servlet](#)
[Học jsp](#)
[Học Hibernate](#)
[Học Struts2](#)
[Học Spring](#)
[Học SQL](#)

Câu hỏi phỏng vấn

[201 câu hỏi phỏng vấn java](#)
[25 câu hỏi phỏng vấn servlet](#)
[75 câu hỏi phỏng vấn jsp](#)
[52 câu hỏi phỏng vấn Hibernate](#)
[70 câu hỏi phỏng vấn Spring](#)
[57 câu hỏi phỏng vấn SQL](#)

About VietTuts.Vn

Hệ thống bài học trên VietTuts.Vn bao gồm thuyết và thực hành về các công nghệ web. Các bài lý thuyết trên hệ thống V tham khảo và tổng hợp từ các trang hi <http://www.tutorialspoint.com>, <http://do>