Bài tập về nhà buổi 3

bài 1)

Cần có tổng 200.000đ từ 3 loại giấy bạc 1000đ, 2000đ, và 5000đ.

Lập chương tình để tìm tất cả các phương án có thể sử dụng các mệnh giá trên để tạo ra tổng 200.000đ

bài 2)

Đổi sang số nhị phân của 1 số nguyên dương n

Ví dụ nhập sao số dương 8 in ra màn hình số nhị phân của nó là 1000

7 -> 111

9 -> 1001

16 -> 10000

Bài 3)

**Đề bài: Kiểm tra và phân loại số trong một dãy**

**Mô tả bài toán:  
Viết chương trình thực hiện các yêu cầu sau trên một dãy số nguyên dương nhập vào từ người dùng:**

1. **Kiểm tra xem số đó có phải là số hoàn hảo hay không.**
2. **Kiểm tra xem số đó có phải là số Armstrong hay không.**
3. **Kiểm tra xem số đó có phải là số đối xứng hay không.**
4. **Tính tổng các chữ số của mỗi số trong dãy.**

**Yêu cầu chi tiết:**

1. **Nhập vào một dãy số nguyên dương n1,n2…nk từ người dùng.**
2. **Với mỗi số trong dãy:**
   * **In ra thông báo số đó là số hoàn hảo hay không.**
   * **In ra thông báo số đó là số Armstrong hay không.**
   * **In ra thông báo số đó là số đối xứng hay không.**
   * **Tính tổng các chữ số và in kết quả.**

**Input:**

* **Một dãy số nguyên dương n1,n2,n3…,nk**
* **Ví dụ: 153 6 121 10 9474**

**Output:**

* **Kiểm tra cho mỗi số trong dãy:**
  + **Kiểm tra số hoàn hảo.**
  + **Kiểm tra số Armstrong.**
  + **Kiểm tra số đối xứng.**
  + **Tổng các chữ số của số đó**

**Ví dụ nhập vào**:

153 6 121 10 9474

**Ví dụ đầu ra**:

153 là số Armstrong.

Tổng các chữ số của 153 là: 9.

6 là số hoàn hảo.

6 là số Armstrong.

6 là số đối xứng.

Tổng các chữ số của 6 là: 6.

121 là số đối xứng.

Tổng các chữ số của 121 là: 4.

10 không phải là số Armstrong.

10 không phải là số hoàn hảo.

10 không phải là số đối xứng.

Tổng các chữ số của 10 là: 1.

9474 là số Armstrong.

9474 là số đối xứng.

Tổng các chữ số của 9474 là: 29.

**Bài 1: Tìm phương án đổi 200.000đ**

## public class Bai1 {

## public static void main(String[] args) {

## for (int soTo1000 = 0; soTo1000 <= 200; soTo1000++) {

## for (int soTo2000 = 0; soTo2000 <= 100; soTo2000++) {

## for (int soTo5000 = 0; soTo5000 <= 40; soTo5000++) {

## if (soTo1000 \* 1000 + soTo2000 \* 2000 + soTo5000 \* 5000 == 200000) {

## System.out.println("1000đ: " + soTo1000 + ", 2000đ: " + soTo2000 + ", 5000đ: " + soTo5000);

## }

## }

## }

## }

## }

## }

**Bài 2: Chuyển số nguyên dương sang nhị phân**

## import java.util.Scanner;

## public class Bai2 {

## public static void main(String[] args) {

## Scanner scanner = new Scanner(System.in);

## System.out.print("Nhập một số nguyên dương: ");

## int n = scanner.nextInt();

## String binary = Integer.toBinaryString(n);

## System.out.println("Số nhị phân của " + n + " là: " + binary);

## }

## }

**Bài 3: Kiểm tra và phân loại số trong một dãy**

## import java.util.Scanner;

## public class Bai3 {

## // Hàm kiểm tra số hoàn hảo

## public static boolean isPerfect(int n) {

## int sum = 0;

## for (int i = 1; i <= n / 2; i++) {

## if (n % i == 0) {

## sum += i;

## }

## }

## return sum == n;

## }

## // Hàm kiểm tra số Armstrong

## public static boolean isArmstrong(int n) {

## int sum = 0, temp = n, digits = String.valueOf(n).length();

## while (temp > 0) {

## int digit = temp % 10;

## sum += Math.pow(digit, digits);

## temp /= 10;

## }

## return sum == n;

## }

## // Hàm kiểm tra số đối xứng

## public static boolean isPalindrome(int n) {

## String str = String.valueOf(n);

## String reversed = new StringBuilder(str).reverse().toString();

## return str.equals(reversed);

## }

## // Hàm tính tổng các chữ số

## public static int sumOfDigits(int n) {

## int sum = 0;

## while (n > 0) {

## sum += n % 10;

## n /= 10;

## }

## return sum;

## }

## public static void main(String[] args) {

## Scanner scanner = new Scanner(System.in);

## System.out.print("Nhập một dãy số nguyên dương (cách nhau bởi dấu cách): ");

## String input = scanner.nextLine();

## String[] numbers = input.split(" ");

## for (String numStr : numbers) {

## int num = Integer.parseInt(numStr);

## System.out.println("\nSố: " + num);

## if (isPerfect(num)) {

## System.out.println(num + " là số hoàn hảo.");

## } else {

## System.out.println(num + " không phải là số hoàn hảo.");

## }

## if (isArmstrong(num)) {

## System.out.println(num + " là số Armstrong.");

## } else {

## System.out.println(num + " không phải là số Armstrong.");

## }

## if (isPalindrome(num)) {

## System.out.println(num + " là số đối xứng.");

## } else {

## System.out.println(num + " không phải là số đối xứng.");

## }

## System.out.println("Tổng các chữ số của " + num + " là: " + sumOfDigits(num));

## }

## }

## }