I have run below JAVA program. When I choose "l" by typing "l" but the program has not exited

Enter n: 2

Enter value of number 1: 2

Enter value of number 2: 5

b. Sum of positive integers

c. Sum of odd integers

d. Number of positive integers

e. Find min integer

f. Find min positive integer

g. Find the last even integer

h. Find the first even integer

i. Find the first prime number

j. Find the last positive integer

k. Find the min even number

l. Exit

Enter your choice: l

Exiting program.

Enter your choice: l

Exiting program.

Enter your choice: l

Exiting program.

Enter your choice:

# BEGIN Baitapbuoi7.java

package Baitapbuoi7;  
  
import java.util.Scanner;  
  
/\* Cho người dùng nhập vào n số nguyên và lưu trữ trong mảng  
 a. Làm menu cho người dùng chọn thực hiện các chức năng bên dưới.  
 b. Tổng các số dương.  
 c. Tính tổng các số lẻ.  
 d. Đếm có bao nhiêu số dương.  
 e. Tìm số nhỏ nhất trong mảng.  
 f. Tìm số dương nhỏ nhất trong mảng  
 g. Tìm số chẵn cuối cùng trong mảng 1 chiều các số nguyên. Nếu mảng không có giá trị chẵn thì trả về -1  
 h. Viết hàm tìm số chẵn đầu tiên trong mảng các số nguyên. Nếu mảng không có giá trị chẵn thì trả về -1  
 i. Tìm số nguyên tố đầu tiên trong mảng 1 chiều các số nguyên. Nếu mảng không có số nguyên tố thì trả về – 1  
 j. Tìm số dương cuối cùng trong mảng số thực. Nếu mảng không có giá trị dương thì trả về -1  
 k. Hãy tìm giá trị chẵn nhỏ nhất trong mảng 1 chiều các số nguyên. Nếu mảng không có số chẵn thì trả về -1  
\*/  
public class Baitapbuoi7 {  
 public static void main(String[] args) {  
 int result;  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter n: ");  
 int n = scanner.nextInt();  
 int[] arr = new int[n];  
 for (int i = 0; i < arr.length; i++) {  
 System.*out*.print("Enter value of number " + (i + 1) + ": ");  
 arr[i] = scanner.nextInt();  
 }  
 scanner.nextLine(); // consume the left new line character in buffer  
 System.*out*.println("b. Sum of positive integers");  
 System.*out*.println("c. Sum of odd integers");  
 System.*out*.println("d. Number of positive integers");  
 System.*out*.println("e. Find min integer");  
 System.*out*.println("f. Find min positive integer");  
 System.*out*.println("g. Find the last even integer");  
 System.*out*.println("h. Find the first even integer");  
 System.*out*.println("i. Find the first prime number");  
 System.*out*.println("j. Find the last positive integer");  
 System.*out*.println("k. Find the min even number");  
 System.*out*.println("l. Exit");  
 String choice = "";  
 while (choice != "l") {  
 System.*out*.print("Enter your choice: ");  
 choice = scanner.nextLine();  
 switch (choice) {  
 case "b": // b. Tổng các số dương  
 System.*out*.println("Sum of positive integers: " + *sumPosInt*(arr));  
 break;  
 case "c": // c. Tính tổng các số lẻ.  
 System.*out*.println("Sum of odd integers: " + *sumOddInt*(arr));  
 break;  
 case "d": // d. Đếm có bao nhiêu số dương.  
 System.*out*.println("Number of positive integers: " + *countPosInt*(arr));  
 break;  
 case "e": // e. Tìm số nhỏ nhất trong mảng.  
 System.*out*.println("Minimum integer: " + *FindMinInt*(arr));  
 break;  
 case "f": // f. Tìm số dương nhỏ nhất trong mảng  
 result = *FindMinPosInt*(arr);  
 if (result == 0) {  
 System.*out*.println("No positive integer found in the list");  
 } else {  
 System.*out*.println("Smallest positive integer: " + result);  
 }  
 break;  
 case "g": // g. Tìm số chẵn cuối cùng trong mảng 1 chiều các số nguyên  
 System.*out*.println("Last even integer: " + *FindLastEvenInt*(arr));  
 break;  
 case "h": // h. Viết hàm tìm số chẵn đầu tiên trong mảng các số nguyên  
 System.*out*.println("First even integer: " + *FindFirstEvenInt*(arr));  
 break;  
 case "i": // i. Tìm số nguyên tố đầu tiên trong mảng 1 chiều các số nguyên.  
 System.*out*.println("First prime number: " + *FindFirstPrimeNum*(arr));  
 break;  
 case "j": // j. Tìm số dương cuối cùng trong mảng số nguyên.  
 System.*out*.println("Last positive integer: " + *FindLastPosInt*(arr));  
 break;  
 case "k": // k. Hãy tìm giá trị chẵn nhỏ nhất trong mảng 1 chiều các số nguyên.  
 System.*out*.println("Smallest even integer: " + *FindMinEvenInt*(arr));  
 break;  
 case "l": // l. Exit  
 System.*out*.println("Exiting program.");  
 break;  
 default:  
 System.*out*.println("Invalid choice. Try again");  
 break;  
 }  
 }  
 }  
  
 public static int sumPosInt(int[] arr) {  
 int sum = 0;  
 for (int num : arr) {  
 if (num > 0) {  
 sum += num;  
 }  
 }  
 return sum;  
 }  
  
 public static int sumOddInt(int[] arr) {  
 int sum = 0;  
 for (int num : arr) {  
 if (num % 2 != 0) {  
 sum += num;  
 }  
 }  
 return sum;  
 }  
  
 public static int countPosInt(int[] arr) {  
 int count = 0;  
 for (int num : arr) {  
 if (num > 0) {  
 count++;  
 }  
 }  
 return count;  
 }  
  
 public static int FindMinInt(int[] arr) {  
 int min = arr[0];  
 for (int num : arr) {  
 if (num < min) {  
 min = num;  
 }  
 }  
 return min;  
 }  
  
 public static int FindMinPosInt(int[] arr) {  
 int minPosInt = Integer.*MAX\_VALUE*;  
 for (int num : arr) {  
 if (num < minPosInt) {  
 minPosInt = num;  
 }  
 }  
 return minPosInt;  
 }  
  
 public static int FindLastEvenInt(int[] arr) {  
 int lastEvenInt = -1;  
 for (int num: arr) {  
 if (num % 2 == 0) {  
 lastEvenInt = num;  
 }  
 }  
 return lastEvenInt;  
 }  
  
 public static int FindFirstEvenInt(int[] arr) {  
 int firstEvenInt = -1;  
 for (int num:arr) {  
 if (num % 2 == 0) {  
 firstEvenInt = num;  
 break;  
 }  
 }  
 return firstEvenInt;  
 }  
  
 public static int FindFirstPrimeNum(int[] arr) {  
 int primeNum = -1;  
 for (int divisor:arr) {  
 boolean isPrimeNumDetected = true;  
 for (int dividend = 2; dividend < divisor; dividend++) {  
 if(divisor % dividend == 0) {  
 isPrimeNumDetected = false;  
 break;  
 }  
 primeNum = divisor;  
 }  
 if(isPrimeNumDetected) {  
 break;  
 }  
 }  
 return primeNum;  
 }  
 public static int FindLastPosInt(int[] arr) {  
 int num = -1;  
 for (int i = 0; i < arr.length; i++) {  
 if (arr[i] > 0) {  
 num = arr[i];  
 }  
 }  
 return num;  
 }  
 public static int FindMinEvenInt(int[] arr) {  
 int min = -1;  
 for (int num: arr) {  
 if (num % 2 == 0) {  
 if (num < min) {  
 min = num;  
 }  
 }  
 }  
 return min;  
 }  
}

# BEGIN Baitapbuoi7.java

Ans1:

The issue with the code is in the while loop condition. In Java, you should use the .equals() method to compare strings, not the == operator. The == operator compares the memory address of the two strings, not their content.

Replace this line:

while (choice != "l") {

with:

while (!choice.equals("l")) {

By doing this, the program will correctly exit when the user enters "l" to exit the menu. Here's the updated code:

// ... (Previous code remains the same)  
  
String choice = "";  
 while (!choice.equals("l")) {  
 System.out.print("Enter your choice: ");  
 choice = scanner.nextLine();  
 switch (choice) {  
 // ... (Previous code remains the same)  
 case "l": // l. Exit  
 System.out.println("Exiting program.");  
 break;  
default:  
 System.out.println("Invalid choice. Try again");  
 break;  
 }  
 }  
  
// ... (Previous code remains the same)

With this modification, the program should now exit properly when the user chooses "l" to exit the menu.