CASH REGISTER VERIFICATION:

1.on In a retail store, the cash register system needs to quickly verify whether any two items in a customer's purchase can be combined to match a specific target amount (e.g., a coupon discount or a bundled offer). This helps in applying promo ons efficiently.

```
Input
4
3 5 2 -4
8
Output
 True
```

PROGRAM:

```
import java.util.Scanner;
public class PairSumSimple {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int n = sc.nextInt();
      int[] prices = new int[n];
      for (int i = 0; i < n; i++) {
         prices[i] = sc.nextInt();
      }
      int target = sc.nextInt();
      boolean found = false;
      for (int i = 0; i < n; i++) {
         for (int j = i + 1; j < n; j++) {
            if (prices[i] + prices[j] == target) {
               found = true;
               break;
            }
         }
         if (found) break;
      }
      System.out.println(found);
   }
}
```

STUDENT ID VERIFICATION:

2.In a classroom management system, a teacher needs to verify whether any student ID appears more than once in the attendance list. This helps prevent errors like duplicate registrations or data entry mistakes.

Example

Input

```
5
 3 5 7 3 9
```

Output

```
True.
```

Input

```
4
10 20 30 40
```

Output

```
 False.
```

PROGRAM:

```
import java.util.Scanner;
public class DuplicateChecker {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
    int[] ids = new int[n];
    for (int i = 0; i < n; i++) {
      ids[i] = sc.nextInt();
    }
    boolean hasDuplicates = false;
    for (int i = 0; i < n; i++) {
      for (int j = i + 1; j < n; j++) {
        if (ids[i] == ids[j]) {
          hasDuplicates = true;
          break;
        }
      }
      if (hasDuplicates) break;
    }
    System.out.println(hasDuplicates);
  }
}
```