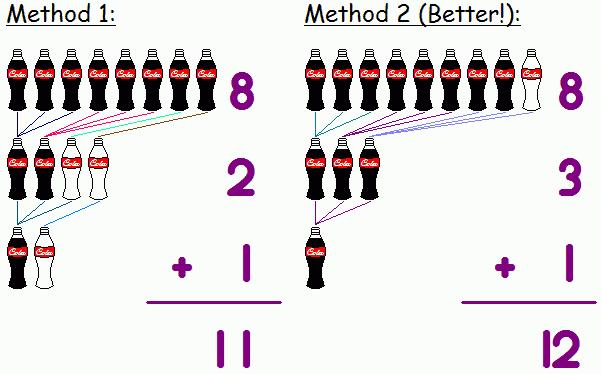
2019/01/04 final

● CPE11150\_cola

便利商店提供了特別的服務:「三個空瓶可以換一瓶可口可樂。」

現在你決定從商店裡買一些(設為N)可口可樂，而想知道最多能拿幾瓶可樂。以下的圖表以N=8 為例。方法一為標準方式:在喝完8瓶可樂之後，有八個空瓶子。拿6個空瓶子換2瓶新的可樂。在喝完全部可樂之後，有4個空瓶子。於是拿3個空瓶換1瓶可樂。最終手中有兩個空瓶，不能再換新的可樂。總共享受用了8 + 2 + 1 = 11瓶可樂。你也可以有更好的方法!方法二:首先先跟朋友借一個空瓶(或是店主)，然後可以享用8 + 3 + 1 = 12瓶可樂!當然，必須將所借的空瓶還給朋友。



○執行畫面範例題型a-單筆 (綠色為輸入，直接剪自eclipse的console)

|  |
| --- |
| ? 8  12 |
| ? 9  13 |
| ? 1  1 |

○執行畫面範例題型b-多筆 (綠色為輸入，直接剪自eclipse的console)

|  |
| --- |
| ? 3  8  9  1  12  13  1 |
|  |

○執行畫面範例題型c-多筆 (綠色為輸入，直接剪自eclipse的console)

|  |
| --- |
| ? 8  9  1  -1  12  13  1 |
|  |

○參考解答

|  |
| --- |
| **import** java.util.Scanner;  **class** CPE11150\_cola\_a {  **static** Scanner *inp* = **new** Scanner(System.***in***);  **public** **static** **void** main(String args[]) {  System.***out***.print("? ");  **int** N = *inp*.nextInt();  **int** ans = *DoIt*(N);  System.***out***.println(ans);  } // ========//  **private** **static** **int** DoIt(**int** N) {  **if** (N <= 1) // 當N<=1時，直接印出目前有幾罐可樂。因為他再怎麼跟朋友借瓶子，也無法歸還空瓶。  **return** N;  **int** ans = N; // ans=最多可以喝多少可樂。初始化為一開始所擁有的可樂數量。  **while** (N != 1) { // 當瓶子剩下1根時，結束迴圈。  **if** (N == 2) // 當N==2時再跟朋友借一個瓶子。  N = N + 1;  ans = ans + N / 3;// 喝掉的可樂數量  N = N % 3 + N / 3;// 目前的瓶子=原本剩下的瓶子+兌換後的瓶子  }  **return** ans;  } // ========//  } /\*\*\*\* end\_of\_class \*\*\*\*/ |
| **import** java.util.Scanner;  **class** CPE11150\_cola\_b {  **static** Scanner *inp* = **new** Scanner(System.***in***);  **public** **static** **void** main(String args[]) {  System.***out***.print("? ");  **int** total = *inp*.nextInt();  **for** (**int** i = 0; i < total; i++) {  **int** N = *inp*.nextInt();  **int** ans = *DoIt*(N);  System.***out***.println(ans);  }  } // ========//  **private** **static** **int** DoIt(**int** N) {  **if** (N <= 1) // 當N<=1時，直接印出目前有幾罐可樂。因為他再怎麼跟朋友借瓶子，也無法歸還空瓶。  **return** N;  **int** ans = N; // ans=最多可以喝多少可樂。初始化為一開始所擁有的可樂數量。  **while** (N != 1) { // 當瓶子剩下1根時，結束迴圈。  **if** (N == 2) // 當N==2時再跟朋友借一個瓶子。  N = N + 1;  ans = ans + N / 3;// 喝掉的可樂數量  N = N % 3 + N / 3;// 目前的瓶子=原本剩下的瓶子+兌換後的瓶子  }  **return** ans;  } // ========//  } /\*\*\*\* end\_of\_class \*\*\*\*/ |
| **import** java.util.Scanner;  **class** CPE11150\_cola\_c {  **static** Scanner *inp* = **new** Scanner(System.***in***);  **public** **static** **void** main(String args[]) {  System.***out***.print("? ");  **while** (**true**) {  **int** N = *inp*.nextInt();  **if** (N < 0)  **break**;  **int** ans = *DoIt*(N);  System.***out***.println(ans);  }  } // ========//  **private** **static** **int** DoIt(**int** N) {  **if** (N <= 1) // 當N<=1時，直接印出目前有幾罐可樂。因為他再怎麼跟朋友借瓶子，也無法歸還空瓶。  **return** N;  **int** ans = N; // ans=最多可以喝多少可樂。初始化為一開始所擁有的可樂數量。  **while** (N != 1) { // 當瓶子剩下1根時，結束迴圈。  **if** (N == 2) // 當N==2時再跟朋友借一個瓶子。  N = N + 1;  ans = ans + N / 3;// 喝掉的可樂數量  N = N % 3 + N / 3;// 目前的瓶子=原本剩下的瓶子+兌換後的瓶子  }  **return** ans;  } // ========//  } /\*\*\*\* end\_of\_class \*\*\*\*/ |