

Instituto Tecnológico de Aguascalientes.

Ingeniería en Tecnologías de la Información y comunicaciones.

Pruebas de software para web. 09:00-10:00 AM. DAB-2104-TC1

Dra. Laura Cecilia Rodríguez Martínez.

Unidad 2.

Actividad 2.1. Ejemplo de surtido de chocolates.

Antonio Santillán Martínez – 20150386

11/03/2024

Contenido

[1.1. Introducción. 1](#_Toc161053543)

[1.2. Desarrollo. 2](#_Toc161053544)

[1.2.1. Particiones. 2](#_Toc161053545)

[1.2.2. Datos de pruebas. 3](#_Toc161053546)

[1.2.3. Programa de las pruebas. 4](#_Toc161053547)

# Introducción.

En este trabajo se verán los siguientes puntos:

Se trata de empacado de chocolates. Con barras de 1 kilo o de 5 kilos. Se calcula el número de barras pequeñas a usar para el empaque de una cierta cantidad de kilos a surtir. Se usan primero las barras grandes y luego las pequeñas.

Entonces una función recibe de entrada la cantidad en existencia para empacar, tanto para barras de 5 como de 1 kilo. La función recibe también la cantidad de kilos a surtir. La función regresa un -1 cuando no se puede surtir la cantidad indicada con la existencia indicada; o regresa el número de empaques pequeños a surtir de acuerdo con la cantidad a surtir y la existencia.

1. Identifique las particiones

1. Plantee dichas particiones
2. Seleccione datos de prueba de entrada para cada partición y su salida esperada

# Desarrollo.

## Particiones.

|  |
| --- |
| **Entradas** |
| Cantidad de barras de 1KG (Ch) |
| Cantidad de Barras de 5KG (G) |
| Cantidad de KG a surtir (C) |

|  |
| --- |
| **Salidas** |
| * ***Numero de barras chicas surtidas en el programa.*** * ***-1 cuando la relación cantidad-existencia no se puede surtir*** |

|  |  |
| --- | --- |
| **Particiones validas** | **Particiones invalidas** |
| (p1) Surtir barras chicas y grandes.  (p2) Surtir solo barras chicas.  (p3) Surtir solo barras grandes.  ***Devolverá el número de barras chicas que se puede surtir si la suma de las entradas Ch y G son igual o mayor al valor de C*** | (p4) Mientras la existencia de las barras sea menor al peso que definimos que se requiera.  (p5) En el caso de que la relación de existencia en barras no permita el surtido sea efectuado.  ***Devolverá -1 cuando sean invalidas*** |

## Datos de pruebas.

Los siguientes datos son los casos de prueba basados en las particiones declaradas en el punto anterior.

Casos de prueba para p1.

|  |  |
| --- | --- |
| Entradas | Salidas |
| |  |  |  | | --- | --- | --- | | Ch | G | C | | 6 | 2 | 15 | | |  | | --- | | Ch | | 5 | |

Casos de prueba para p2.

|  |  |
| --- | --- |
| Entradas | Salidas |
| |  |  |  | | --- | --- | --- | | Ch | G | C | | 5 | 0 | 3 | | 9 | 0 | 3 | | |  | | --- | | Ch | | 3 | | 3 | |

Casos de prueba para p3.

|  |  |
| --- | --- |
| Entradas | Salidas |
| |  |  |  | | --- | --- | --- | | Ch | G | C | | 3 | 1 | 5 | | |  | | --- | | Ch | | 0 | |

Caso de prueba para p4.

|  |  |
| --- | --- |
| Entradas | Salidas |
| |  |  |  | | --- | --- | --- | | Ch | G | C | | 1 | 3 | 17 | | |  | | --- | | -1 | |

Casos de prueba para p5.

|  |  |
| --- | --- |
| Entradas | Salidas |
| |  |  |  | | --- | --- | --- | | Ch | G | C | | 13 | 1 | 18 | | |  | | --- | | -1 | |

## Programa de las pruebas.

1. <?php
2. declare(strict\_types=1);
3. use PHPUnit\Framework\TestCase;
4. final class EjemploChocoExac extends TestCase
5. {
6. public function testp1(): void
7. {
8. $this->assertEquals(5, c(6,2,15));
9. }
10. public function testp21(): void
11. {
12. $this->assertEquals(3, c(5,0,3));
13. }
14. public function testp22(): void
15. {
16. $this->assertEquals(3, c(9,0,3));
17. }
18. public function testp3(): void
19. {
20. $this->assertEquals(0, c(3,1,5));
21. }
22. public function testp4(): void
23. {
24. $this->assertEquals(-1, c(1,3,17));
25. }
26. public function testp5(): void
27. {
28. $this->assertEquals(-1, c(13,1,19));
29. }
30. }
31. function c($cuantasbChica,$cuantasbGrande,$totalaSurtir)
32. {
33. $maxbGrande = $totalaSurtir/5;
34. if($maxbGrande  < $cuantasbGrande){
35. $empacarGrandes  = $cuantasbGrande;
36. }
37. else{
38. $empacarGrandes  = $cuantasbGrande;
39. }
41. $totalaSurtir = $totalaSurtir - ($empacarGrandes\*5);
43. if($cuantasbChica <= $totalaSurtir){
44. return -1;
45. }
46. else{
47. return $totalaSurtir;
48. }
49. }
50. ?>

Donde observamos que ese programa es una representación de lo que pide inicialmente el siguiente párrafo.

---------------------CÓDIGO A PROBAR.

... primero traducir a PHP, exacto como está programado aquí,

... luego probar con el mismo programa de pruebas que ya prepararon

... por último reportar resultados en la misma tabla donde indicaron los casos de prueba ya sea incluso por requerimiento o al menos por particiones.

public class ChocolateEmpacado {

public int calculoBarras

( int cuantasbChica, int cuantasbGrande, int totalaSurtir)

{

int maxbGrande = totalaSurtir / 5;

if (maxbGrande < cuantasbGrande) empacarGrandes = maxbGrande;

else empacarGrandes = cuantasbGrande;

totalaSurtir = totalaSurtir – (empacarGrandes\*5);

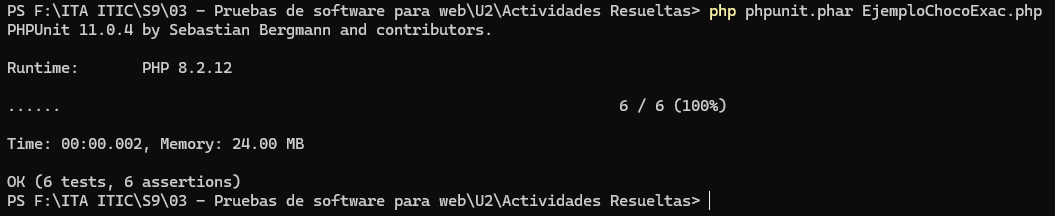
if(cuantasbChica<=totalaSurtir) return -1;

else return totalaSurtir;

           }

}

Siendo su exacto en PHP. Donde al realizar las pruebas que tenemos en las particiones nos da lo siguiente:



Que nos menciona que los datos de prueba que declaramos en las particiones fueron completadas y todo paso sin problema.