# **Gants**

**Fait travailler** : *(ex :boucles, conditions, POO...)*

* boucles
* arrondis

**Enoncé du Dojo :**

Il faut déterminer combien de paires de gants de la même couleurs sont dans un sac. Le sac est représenté par un tableau en entrée, et il faut renvoyer un integer en sortie.

**Exemples** :

Entrée [‘vert’, ‘bleu’, ‘vert’, ‘rouge’] Résultat = 1 (1 paire de gants vert, le reste sont des gants isolés donc ne formant pa s une paire)

Entrée [‘jaune, ‘jaune, ‘vert’, ‘vert’, ‘jaune’, ‘bleu’] => 2 (1 paire complète de jaunes et 1 de vert, le reste sont des gants isolés)

**Correction :**

**class** Gant

{

**public function** nbPaires($stock)

{

$gants = [];

**foreach** ($stock **as** $couleur) {

**if** (*array\_key\_exists*($couleur, $gants)) {

$gants[$couleur]++;

} **else** {

*// ajoute la couleur dans $gants et l'initialise à 1*

$gants[$couleur] = 1;

}

}

$nbPaire = 0;

**foreach** ($gants **as** $nbGant) {

*// on divise par 2 le nombre de gants par couleur, on arrondit a l'entier inferieur*

*// on ajoute le nombre de paires obtenu a $nbpaire*

$nbPaire += *floor*($nbGant / 2);

}

**return** $nbPaire;

}

}

**Tests :**

**class** GantTest **extends** TestCase

{

**public function** testPaire ()

{

$gant = **new** Gant;

$this->assertequals(0, $gant->nbPaires([**'vert'**]));

$this->assertequals(1, $gant->nbPaires([**'vert'**,**'vert'**]));

$this->assertequals(1, $gant->nbPaires([**'vert'**,**'vert'**,**'bleu'**]));

$this->assertequals(0, $gant->nbPaires([**'vert'**,**'rouge'**,**'bleu'**]));

$this->assertequals(0, $gant->nbPaires([]));

$this->assertequals(2, $gant->nbPaires([**'vert'**,**'bleu'**,**'bleu'**,**'vert'**]));

}

}