

Moyon Xavier
Radenne Gauthier
Vasset Nathan

Jeux d'essai

InitialiserCarreSimple :

_Pour cette fonction il suffit de faire un
assertEquals("ABCDEFGHJKLMNOPQRSTUVWXYZ"), cela permettra de vérifier si le carré est
valable

AfficherCarre :

_Pour cette fonction il faut la vérifier sans assert étant donné qu'il s'agit d'une
fonction d'affichage

coderLettre :

_Pour cette fonction on teste les extremes tel que A pour le debut de ligne et colonne
et Z pour la fin de ligne et de colonne.
On verifie aussi si la fonction traite bien l'exception du W.

```
_assertEquals(coderLettre("ABCDEFGHJKLMNOPQRSTUVWXYZ", 'A'), "00");  
_assertEquals(coderLettre("ABCDEFGHJKLMNOPQRSTUVWXYZ", 'Z'), "44");  
_assertEquals(coderLettre("ABCDEFGHJKLMNOPQRSTUVWXYZ", 'C'), "02");  
_assertNotEquals(coderLettre("ABCDEFGHJKLMNOPQRSTUVWXYZ", 'Z'), "55");  
_assertEquals(coderLettre("ABCDEFGHJKLMNOPQRSTUVWXYZ", 'W'), "41");
```

coderMessage :

_On verifie si la fonction retourne le code avec la bonne norme ou si la fonction
fonctionne tout simplement :

```
_assertEquals(coderMessage("ABCDEFGHJKLMNOPQRSTUVWXYZ", "BUTINFORM  
ATIQUE"), "01 40 34 13 23 10 24 32 22 00 34 13 31 40 04 "); lci on  
verifie si la fonction fonctionne
```

```
_assertEquals(coderMessage("ABCDEFGHJKLMNOPQRSTUVWXYZ", "AEKOUZ"),  
"00 04 20 24 40 44 "); lci on verifie si la fonction retourne le code sous la  
bonne norme
```

```
_assertNotEquals(coderMessage("ABCDEFGHJKLMNOPQRSTUVWXYZ", "AEKOUZ"), "00 04 20 24 40 44");
```

decoderLettre :

Ici on test les extremes et l'exception pour W qui doit retourner V et on test avec différents carrés.

```
_assertEquals(decoderLettre("ABCDEFGHJKLMNOPQRSTUVWXYZ", "00"), 'A') ; Ici on test l'extreme pour les lignes et les colonnes
```

```
_assertEquals(decoderLettre("ABCDEFGHJKLMNOPQRSTUVWXYZ", "04"), 'E') ; Ici on test l'extreme pour les colonnes
```

```
_assertEquals(decoderLettre("ABCDEFGHJKLMNOPQRSTUVWXYZ", "40"), 'U') ; Ici on test l'extreme pour les lignes
```

```
_assertEquals(decoderLettre("ABCDEFGHJKLMNOPQRSTUVWXYZ", "44"), 'Z') ; Ici on test l'extreme pour les lignes et les colonnes
```

```
_assertEquals(decoderLettre("ABCDEFGHJKLMNOPQRSTUVWXYZ", "32"), 'R') ; Ici on test une valeur aleatoire
```

```
_assertEquals(decoderLettre("BACDEFGHJKLMNOPQRSTUVWXYZ", "00"), 'B') ; Ici on test une valeur aleatoire
```

```
_assertEquals(decoderLettre("ABCDEFGHJKLMNOPQRSTUVWXYZ", "41"), 'V') ; Ici test la valeur pour V ou W
```

decoderMessage :

estPresent :

Ici on verifie la fonction avec differentes valeurs aléatoire pour voir si elles fonctionne bien.

```
_assertTrue(estPresent("BONJOUR", 'B')) ;
```

```
_assertTrue(estPresent("BUTINFO", 'I')) ;
```

```

    _assertTrue(estPresent("BONJOUR", 'O'));

    _assertFalse(estPresent("BONJOUR", 'M'));

    _assertFalse(estPresent("BONJOUR", '0'));

    _assertFalse(estPresent("SAE", 'O'));

    _assertFalse(estPresent("IUTA", 'M'));

```

initialiserCarreAvecCle :

_On test la fonction avec une chaîne vide qui devra nous retourner l'alphabet sans W de même si on met juste W en parametre ensuite on test avec un mot contenant W et avec un contenant des doublons

```

    _assertEquals("BUTINFORMAQECDGHJKLPSVXYZ", initialiserCarreAvecCle("BUTINFORMATIQUE"));

    _assertEquals("ABCDEFGHJKLMNOPQRSTUVWXYZ", initialiserCarreAvecCle(""));

    _assertEquals("ABCDEFGHJKLMNOPQRSTUVWXYZ", initialiserCarreAvecCle("W"));

    _assertEquals("AGONBCDEFHIJKLMPQRSTUVWXYZ", initialiserCarreAvecCle("WAGON"));

    _assertEquals("GROUPEABCFHIJKLMNQSTVXYZ", initialiserCarreAvecCle("GROUPEE"));
}

```

estLettreMajuscule :

_Pour cette fonction on test la fonction avec des valeurs aléatoires tel que des lettres en majuscules et en minuscules et ensuite avec des caractere differents qui ne sont pas de lettres

```

    _assertTrue(estLettreMajuscule('A'));
    _assertTrue(estLettreMajuscule('Z'));
    _assertTrue(estLettreMajuscule('B'));
    _assertTrue(estLettreMajuscule('V'));

    _assertFalse(estLettreMajuscule('a'));
    _assertFalse(estLettreMajuscule('z'));

```

```

_assertFalse(estLettreMajuscule('b'));
_assertFalse(estLettreMajuscule('v'));

_assertFalse(estLettreMajuscule(' '));
_assertFalse(estLettreMajuscule('!'));

_assertFalse(estLettreMajuscule('0'));
_assertFalse(estLettreMajuscule('9'));

```

estCleValide :

_Ici on verifie que la fonction fonctionne en utilisant des espaces des miniscules et des caracteres autre que des lettres

```

_assertTrue(estCleValide("BUTINFORMATIQUE"));
_assertFalse(estCleValide("BUTINF ORMATIQUE"));
_assertFalse(estCleValide("BUTINFORMATIQUE!"));
_assertFalse(estCleValide("ButInformatique"));
_assertFalse(estCleValide(" BUTINFORMATIQUE"));
_assertFalse(estCleValide("BUTINFORMATIQUE "));

```

estChiffreOK :

_Ici on verifie que la fonction fonctionne avec des valeurs extremes tels que -1 , LARGEUR et 0 et LARGEUR-1

```

_assertTrue(estChiffreOK(0));
_assertTrue(estChiffreOK(LARGEUR-1));
_assertFalse(estChiffreOK(-1));
_assertFalse(estChiffreOK(LARGEUR));

```

estMessageCodeValide :

_Ici on verifie si la fonction retourne bien un code avec les bonnes normes et qu'elles n'accepte pas les valeurs trop grandes

```

_assertTrue(estMessageCodeValide("21 00 01 "));
_assertTrue(estMessageCodeValide("34 44 21 22 "));
_assertFalse(estMessageCodeValide("41 4"));
_assertFalse(estMessageCodeValide("45 41 "));

```

estMessageValide :

Ici on verifie si la fonction avec des caracteres autres que des lettres tel que des numeros et des espaces ensuite on verifie avec des lettres minuscules et majuscules et une chaîne vide.

```
_assertTrue(estMessageValide("BUTINFORMATIQUE"));
_assertTrue(estMessageValide("GROUPEW"));
_assertFalse(estMessageValide("BUT INFORMATIQUE"));
_assertFalse(estMessageValide("BUTinforMATIQUE"));
_assertFalse(estMessageValide("BUTINFORMATIQUE4"));
```