

Dans cet exercice, toutes les images sont :

- composées de  $5 \times 5$  pixels ;
- en noir et blanc.

Chaque groupe de 3 élèves doit avoir :

- cet énoncé ;
- les exemples d'images ;
- une feuille de papier A4 quadrillée.

## 1 Transmission d'image

**But** Trouver comment encoder une image.

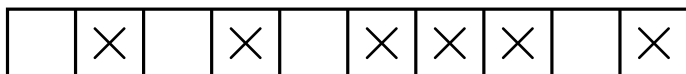
1. Distribuer les rôles entre les trois membres du groupe :

**Émetteur** Il envoie l'image.

**Récepteur** Il reçoit l'image.

**Arbitre** Il vérifie que les règles sont respectées.

2. Trouver une manière de coder l'image, sachant que la seule information que peut transmettre l'émetteur est une ligne d'une feuille A4, dont certaines cases ont été marquées, comme dans l'exemple suivant.

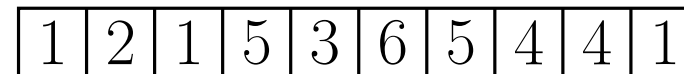


3. Une fois que le groupe s'est mis d'accord sur une méthode :
  - (a) l'émetteur prend une image dans la liste d'exemples (que le récepteur ne voit pas) ;
  - (b) il l'encode sur une ligne de la feuille ;
  - (c) il transmet cette feuille au récepteur ;
  - (d) le récepteur reproduit l'image.
  - (e) L'arbitre vérifie que les règles sont respectées (en particulier, que l'émetteur et le récepteur ne communiquent rien d'autre que cette bande de papier).

## 2 Compression

**But** Transmettre une image, en utilisant le moins de caractères possibles.

1. L'exercice est le même que le précédent, sauf que cette fois-ci, les cases de la ligne ne contiennent plus des croix, mais des chiffres de 1 à 6, comme sur l'exemple suivant.

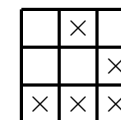


2. Refaire la manipulation 3 de la partie 1. Le but est d'utiliser le moins de cases possibles.
3. Une fois que votre code fonctionne, l'expliquer à un autre groupe. L'autre groupe vous explique le sien. Votre but est alors de construire :
  - (a) une image qui ne puisse pas être représentée par ce code (ce n'est pas forcément possible) ;
  - (b) une image qui utilise le plus de cases possibles.

## 3 Intégrité

**But** Transmettre une image, et être capable de déceler si l'image a été altérée.

Pour cet exercice, le message transmis utilise toujours des cases, marquées ou non, mais peut utiliser deux dimensions, comme dans l'exemple ci-contre.



1. Inventer une manière de coder l'image de la même manière qu'à la partie 1, sauf qu'il faut que le récepteur soit capable de dire si le message a été altéré ou non.
2. Il n'y a plus d'arbitre, mais un transmetteur :
 

**Émetteur** Il code l'image, et la donne au transmetteur ;

**Transmetteur** Il modifie (ou non), une seule case du message ;

**Récepteur** Il reçoit le message, dit si l'image a été altérée ou non, avant de la reconstituer.
3. Trouver un algorithme qui permette au récepteur de savoir si l'image a été altérée ou non.
4. Échanger votre algorithme avec un autre groupe ; essayer de trouver un bug, par exemple :
  - une image qui ne puisse pas être transmise avec cette méthode ;
  - une image telle que le récepteur pense qu'elle a été altérée, à tort ;
  - une image telle que le récepteur pense qu'elle n'a pas été altérée, à tort.
5. Si l'autre groupe a trouvé un bug, le corriger.

## 4 Correction

**But** Transmettre une image, et être capable de corriger l'image si elle a été altérée.

Les modalités sont les mêmes qu'à l'exercice précédent, sauf que si le message a été altéré, le récepteur doit être capable de corriger l'erreur.