

1 Permutations

On utilisera CaRMetal pour programmer en javascript. Dans le menu **Javascript>Nouveau script dans la construction** ouvre l'éditeur javascript.

Construire la permutation identité des n premiers entiers $1, \dots, n$. Écrire une procédure qui fabrique une permutation au hasard :

```
function permutationHasard(n){
    var p = new Array(n);
    var d = 0;
    for(var i=0; i<n; i++){
        d = Math.floor(Math.random() * (i + 1));
        p[i] = p[d];
        p[d] = i+1;
    }
    return p;
}
```

Construire une permutation. La visualiser avec `Alert(p)`; et exécuter le programme en cliquant sur le feu tricolore. Arrêter une boucle infinie en cliquant sur la flèche tournante à gauche du feu tricolore.

Écrire un prédicat testant si un tableau est bien une permutation. Tester si la permutation construite en est bien une sinon `Alert(p+' n'est pas une permutation')`; Construire à la main un mauvais tableau.

Afficher une permutation sous la forme standard en utilisant l'environnement `pmatrix` :

```
\begin{pmatrix}
a & b & c \\
d & e & f
\end{pmatrix}
```

Les chaînes de caractères se concatènent avec l'opérateur `+` et on peut attacher le texte à la place d'un point comme suit :

```
var A = Point();
SetShowName(A, true); // Pour afficher le texte
SetAlias(A, "$\\begin{pmatrix}"+tid+" \\ \\ "+ tpe+"\\end{pmatrix}$");
```

On peut renvoyer ce point de manière à faire `Move(affiche(p), 2, 3)`; pour bouger l'affichage au point de coordonnées (2,3).

Construire la composée de deux permutations puis l'utiliser pour calculer la puissance d'une permutation. Faire afficher les premières puissances de la permutation aléatoire `p` du début.

Construire l'orbite d'un élément. Adapter la procédure pour retourner le tableau des cycles à supports disjoints composant la permutation. Permettre l'affichage de ces cycles.

L'ordre d'une permutation est le plus petit commun multiple de la longueur de ses cycles. Afficher les puissances de la permutation jusqu'à son ordre.

Une permutation peut s'écrire comme un produit de transpositions. Programmer la procédure correspondante ainsi que la décomposition en produit de transpositions élémentaires. Utiliser cette décomposition pour visualiser une transposition élémentaire puis les composer pour visualiser une permutation (du haut vers le bas) à l'aide de

```
for(var i=1; i<k; i++)
    Segment(Point(i,-h),Point(i,-(h+1)));
```

Trouver un algorithme permettant de calculer la permutation suivante dans l'ordre lexicographique.

2 Anthyphérèse

L'algorithme d'Euclide itéré permet de calculer le plus grand commun diviseur de deux nombres. Utiliser le tableau de Géogébra pour visualiser ce processus comme le retranchement de carrés de côté b d'un rectangle de côtés $a \times b$.

