

Chez les papous, il y a des papous papas, des papous pas papas, des pas papous papas et des pas papous pas papas. Mais certain ont des poux... Nous avons donc les papous papas à poux, des papous papas pas à poux, des papous pas papas à poux, ...

Ce projet est la suite du projet Gaston. On avait en effet construit un algorithme récursif pour écrire la liste de toutes les catégories de personnes. Dans ce projet, on cherche à dé-récursiver l'algorithme.

On rappelle qu'une pile est une structure de donnée qui enregistre des informations selon le mode dernier entré premier sorti (LIFO : Last In First Out). On manipule une pile en utilisant les quatres opérations suivantes :

Création d'une nouvelle pile	PileVide() : Pile	
Teste si la pile est vide	EstVide(Pile) : Booleen	
Ajout d'un élément	Empiler(T , Pile)	modifie la pile
Suppression d'un élément	Depiler(Pile) : T	modifie la pile

Note : Depiler(p) est valide seulement si non EstVide(p).

On rappelle également que pour implanter la récursivité le compilateur utilise la pile d'exécution réservée par le système d'exploitation dans le cas du C/C++ ou la pile de la machine virtuelle dans le cas de Java. En utilisant une pile que l'on a construit soit même, on peut donner une version non récursive de tous les algorithmes. Voici comment faire sur le problème de Gaston : On utilise une pile dans laquelle on va stocké les paramètres des appels récursifs, par exemple des couples (choix, int) où l'entier contient le nombre de caractéristiques qui sont fixées dans choix. Le code s'écrit alors sous la forme d'une boucle où l'on dépile le travaille pour simuler l'entrée dans une fonction et on empile pour simulé l'appelle de la fonction :

```
pile p = PileVide()
p.empile((choix vide initial, 0))
Tant que non p.est_vide() faire
  choix, ichoix <- p.depile()
  si ichoix == nombre de caractéristique alors
    // toute les caractéristique ont été fixée dans choix
    afficher choix
  sinon
    empiler (choix + caractéristique[ichoix],      ichoix+1)
    empiler (choix + "pas" + caractéristique[ichoix], ichoix+1)
```

1. Donner une implantation de la pile par tableau où toutes les opérations (sauf peut-être la construction) ont un coût $O(1)$.
2. Donner une implantation de la pile par liste chaînée où toutes les opérations ont un coût $O(1)$. ;
3. En utilisant l'algorithme ci dessus, résoudre de manière non récursive le problème de Gaston. La solution doit être capable de fonctionner avec les deux implantations de la pile.