

Devoir #3 : récursivité

I. n -chaînes

On appelle n -**chaîne** une chaîne de caractères composée uniquement de chiffres (0 – 9). Les chiffres dans une n -chaîne doivent toujours être en ordre croissant et peuvent se répéter. Voici quelques exemples valides de 10-**chaînes** :

```
0000000000
0000335666
1234567899
9999999999
8888899999
```

En vous inspirant en particulier des exemples `genere_1.cpp` et `ligne_sudoku.cpp` qui se trouvent dans le groupe Teams “INFO4302 01M Hiver 2023” dans le dossier Exercices/, **écrivez un programme récursif qui prend en entrée un entier n et qui génère toutes les n -chaînes valides.**

Exemple d'entrée :

```
2
```

Exemple de sortie :

```
00
01
02
03
04
05
06
07
08
09
11
12
13
14
15
16
17
```

18
19
22
23
24
25
26
27
28
29
33
34
35
36
37
38
39
44
45
46
47
48
49
55
56
57
58
59
66
67
68
69
77
78
79
88
89
99

II. Le langage SBM

La Société Binaire de Moncton (SBM) utilise un langage secret binaire où toutes les séquences de bits peuvent s'écrire $a_1a_2\dots a_n$ ($n \geq 1$) où chaque a_i vaut 0, 01 ou 11. Par exemple, les séquences 001 et 01111 sont des séquences SBM valides, mais pas 111 ni 1110.

Vous devez écrire un programme récursif qui prend en entrée un entier T indiquant le nombre de séquences à tester. Puis sur chacune des T lignes suivantes, une séquence binaire. Pour chaque séquence, votre programme doit afficher sur une ligne séparée OUI ou NON selon que la séquence appartient au langage SBM ou non.

Exemple d'entrée :

```
7
11001
01111
111
1110
0
100000
11010
```

Exemple de sortie :

```
OUI
OUI
NON
NON
OUI
NON
OUI
```