

BTS SIO

Epreuve d'algorithmique – U22

Contrôle en cours de formation

Déroulement de l'épreuve :

Cette épreuve de Contrôle en Cours de Formation (CCF) se déroule en trois étapes :

- **Etape 1 : Ecrit (30 min au maximum).**

Vous devez traiter la partie A du sujet.

Vous inscrirez les réponses dans le « document réponse » page 5 et page 6 de ce sujet.

Les machines (ordinateur, calculatrice, ..) sont interdites pour cette partie.

Ce document réponse est à rendre au surveillant de la salle avant de passer à la seconde étape.

- **Etape 2 : Sur machine (30 min au minimum).**

Vous devez traiter la partie B du sujet à l'aide d'un ordinateur.

Au moment de la remise du « document réponse » de l'étape 1 le surveillant vous indiquera l'ordinateur à utiliser.

La durée totale pour effectuer les deux premières étapes est d'exactly une heure.

- **Etape 3 : Oral (20 min).**

L'examineur est votre professeur d'algorithmique appliquée.

Cette partie se déroule en deux temps. Tout d'abord, vous disposez de 10 minutes pour présenter votre travail de l'étape 2, puis au cours des 10 minutes suivantes un entretien permet de préciser votre démarche.

Ce sujet comporte 7 pages

SUDOKU 81 cases

Présentation du sujet :

Le Sudoku est un jeu consistant à compléter une grille. Le joueur dispose d'une grille de 81 cases organisées en 9 lignes de 9 cases. Les 81 cases sont par ailleurs divisées en 9 carrés de 9 cases. (cf. dessin ci-dessous).

Le joueur dispose d'une grille partiellement remplie et doit compléter les chiffres manquants en respectant les règles suivantes:

- Chaque case contient un chiffre compris entre 1 et 9.
- Chaque ligne contient tous les chiffres de 1 à 9. Chaque chiffre apparaît donc une fois et une seule par ligne.
- Chaque colonne contient tous les chiffres de 1 à 9. Chaque chiffre apparaît donc une fois et une seule par colonne.
- Chaque carré contient tous les chiffres de 1 à 9. Chaque chiffre apparaît donc une fois et une seule par carré.

La grille ci-dessous est dite conforme, c'est à dire complète, avec les règles du remplissage respectées.

1	2	3	4	5	6	7	8	9	→ ligne
7	8	9	1	2	3	4	5	6	
4	5	6	7	8	9	1	2	3	
9	1	2	3	4	5	6	7	8	
6	7	8	9	1	2	3	4	5	
3	4	5	6	7	8	9	1	2	
8	9	1	2	3	4	5	6	7	
5	6	7	8	9	1	2	3	4	→ carré
2	3	4	5	6	7	8	9	1	
									↓ colonne

On se propose d'écrire un algorithme qui permet de contrôler la conformité des grilles remplies par un joueur.

On stocke la grille de Sudoku dans une matrice de nombres de 9 lignes, 9 colonnes numérotées de 0 à 8 de la manière suivante :

	0	1	2	3	4	5	6	7	8
0									
1									
2									
3									
4									
5									
6									
7									
8									

PARTIE A :**Utilisation d'un ordinateur interdite. Les réponses sont à rédiger sur les feuilles réponses****Durée : 30 minutes**

On considère la procédure ci-dessous.

procédure sudotestline(A: tableau):

variables:

i, testlin: entiers

début

testlin ← 0

pour i de 0 à 8

testlin ← testlin*A[i]

si testlin ≠ 362880 alors

afficher ("Il y a un problème dans la ligne")

fin de si

fin de pour

fin de procédure

- 1) On lance cette procédure avec le tableau $A=[1,2,3,1,2,3,1,1,1]$. Que fait la procédure **sudotestline** ?
- 2) On lance la procédure avec le tableau $A=[1,2,3,4,5,6,7,8,9]$. Que fait la procédure **sudotestline** ?
- 3) Quel(s) problème(s) constatez-vous ? Proposer une correction de la procédure afin qu'elle calcule bien le produit des valeurs du tableau avant le test.
- 4) Sachant que $9! = 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 362880$, expliquer en quoi cette procédure permet de tester une ligne d'une grille de Sudoku.
- 5) On a modifié, ci-dessous, la procédure **sudotestline**, en ajoutant une boucle, afin qu'elle teste toutes les lignes d'un tableau A de 9 lignes et 9 colonnes (On cherche les lignes ne contenant pas tous les chiffres de 1 à 9, une et une seule fois).
 - a) Cette procédure ne fonctionne pas, car l'initialisation de la variable **testlin** a été oubliée.
Ajouter l'initialisation de la variable **testlin**.
 - b) Compléter la procédure afin qu'elle affiche, le message "Pas de problème sur les lignes", si toutes les lignes sont correctes.

procédure sudotestline(A : tableau) :

variables:

i, j, testlin : entiers

début

pour i de 0 à 8

pour j de 0 à 8

testlin ← testlin * A[i][j]

fin de pour

si testlin ≠ 362880 alors

afficher ("Il y a un problème dans la ligne", i)

fin de si

fin de pour

fin de procédure

PARTIE B :
Durée : 30 minutes

Pour cette partie, une implémentation de vos solutions algorithmiques est demandée. Pour cela vous devez utiliser un ordinateur.

Toutefois, en cas de difficulté d'implémentation, un algorithme en langage naturel sera pris en compte dans l'évaluation. Dans ce cas seulement, vous rédigerez votre algorithme sur la « feuille réponse » de la page 7.

Vous enregistrerez votre travail sur une clé USB dans un dossier à votre nom.

Pensez à sauvegarder régulièrement votre travail.

Vous trouverez sur le bureau de l'ordinateur qui vous a été désigné un dossier à votre nom dans lequel se trouve un fichier intitulé : SUDOKU à compléter.py.

Commencez par enregistrer ce fichier sous **la clef USB**

Le code partiel, contient la procédure **sudotestline(A)** de la partie A, la procédure **affsudok(A)** qui permet l'affichage de la grille de Sudoku, ainsi que la procédure **sudotestcarre(A)** qui teste les carrés de la grille.

L'objet de cette partie est de compléter le code de la procédure **sudotestcol(A)** décrite à la question 1, d'écrire une procédure **sudotestinterval(A)** décrite question 2 et d'ajouter des instructions nécessaires aux contrôles des grilles proposées cf. question 3.

Question 1 :

Compléter la procédure **sudotestcol(A)**, qui teste les colonnes de la grille. C'est à dire, qui vérifie toutes les colonnes du tableau A de 9 lignes et 9 colonnes. Si tous les chiffres de 1 à 9 n'apparaissent pas une et une seule fois par colonne, elle indique les colonnes qui posent problème. Si toutes les colonnes sont correctes, elle affiche le message : "Pas de problème sur les colonnes".

Question 2 :

Ecrire une procédure **sudotestinterval(A)**, qui teste les valeurs de la grille afin de déterminer les valeurs non comprises entre 1 et 9. Cette procédure affichera la ligne et la colonne des valeurs hors intervalle.

Question 3 :

Compléter le code afin d'afficher et de tester les grilles de Sudoku proposées. Le code retournera le message "grille conforme" si la grille testée est conforme ou "grille non conforme" si elle ne l'est pas, en indiquant la position des éventuels problèmes rencontrés. Vous utiliserez les procédures fournies par l'énoncé et celles que vous avez écrites à la question 1 et à la question 2.

FEUILLE REPONSE DE LA PARTIE A

1) On lance cette procédure avec le tableau $A = [1, 2, 3, 1, 2, 3, 1, 1, 1]$.

Que fait la procédure **sudotestline** ?

2) On lance la procédure avec le tableau $A = [1, 2, 3, 4, 5, 6, 7, 8, 9]$.

Que fait la procédure **sudotestline** ?

3) Quel(s) problème(s) constatez-vous ?

Proposer une correction de la procédure afin qu'elle calcule bien le produit des valeurs du tableau avant le test.

procedure sudotestline(A : tableau) :

variables:

i , testlin : entiers

début

testlin \leftarrow 0

pour i de 0 à 8

testlin \leftarrow testlin * A[i]

si testlin \neq 362880 alors

afficher ("Il y a un problème dans la ligne")

fin de si

fin de pour

fin de procedure

4) Sachant que $9! = 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 362880$, expliquez en quoi cette procédure permet de tester une ligne d'une grille de Sudoku.

5) On a modifié, ci-dessous, la procédure **sudotestline**, en ajoutant une boucle, afin qu'elle teste toutes les lignes d'un tableau A de 9 lignes et 9 colonnes (On cherche les lignes ne contenant pas tous les chiffres de 1 à 9, une et une seule fois).

- a) Cette procédure ne fonctionne pas, car l'initialisation de la variable **testlin** a été oubliée.
Ajouter l'initialisation de la variable **testlin**.
- b) Compléter la procédure afin qu'elle affiche, le message "Pas de problème sur les lignes", si toutes les lignes sont correctes.

```
procedure sudotestline(A: tableau):
```

```
  variables:
```

```
    i , j, testlin : entiers
```

```
  début
```

```
    pour i de 0 à 8
```

```
      pour j de 0 à 8
```

```
        testlin ← testlin * A[i][j]
```

```
      fin de pour
```

```
      si testlin ≠ 362880 alors
```

```
        afficher ("Il y a un problème dans la ligne", i)
```

```
      fin de si
```

```
    fin de pour
```

```
  fin de procedure
```

NOM et Prénom :

Heure de passage :

FEUILLE REPONSE DE LA PARTIE B

Cette feuille réponse ne doit être utilisée que si vous ne parvenez pas à implémenter vos réponses aux questions de la partie B. Dans ce cas, rédigez vos algorithmes en langage naturel ci-dessous.