



Maturité gymnasiale

Session 2019

## EXAMEN DE L'OPTION COMPLÉMENTAIRE INFORMATIQUE

### Informations et consignes :

- Temps à disposition : 3 heures.
- Aucun document n'est autorisé.
- Calculatrice non programmable autorisée.
- Le nombre de points est indiqué pour chacun des cinq problèmes. Il y a 100 points au total.
- Écrire proprement au stylo ou à l'encre. La présentation est prise en compte pour la note.
- Utilisez une feuille par problème.
- Écrivez votre nom sur chaque feuille.
- Faites une marge de 2 cm à gauche.
- Rendez tous vos documents, y compris la donnée.

### Problème 1 : système binaire (15 points)

Certains logiciels utilisent la représentation POSIX du temps, dans laquelle le temps est représenté comme un nombre de secondes écoulées depuis le 1<sup>er</sup> janvier 1970 à minuit (0 heure). Sur les ordinateurs 32 bits, la plupart des systèmes d'exploitation représentent ce nombre comme un **nombre entier signé de 32 bits**.

- a) Quel est le nombre de secondes maximum que l'on peut représenter ?
- b) À quelle date cela correspond-il (jour, mois, année, heures, minutes, secondes).  
*Indications :*
  - 1) *afin de tenir compte des années bissextiles, comptez par cycles de 4 ans composés de  $4 \cdot 365 + 1 = 1461$  jours ;*
  - 2) *l'an 2000 est une année bissextile.*
- c) Que se passera-t-il une seconde plus tard ? Quel sera le nombre de secondes affiché (en base 10) ?

### Problème 2 : structure de données (20 points)

On utilise un tas où l'élément le plus petit est à la racine.

- a. Entrer les nombres suivants, dans cet ordre, dans un tas. Dessiner l'évolution du tas, sous forme d'un arbre, étape par étape.

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

- b. On retire l'élément se trouvant à la racine du tas. Dessiner le tas après cette opération.  
c. À quoi peut servir un tas ? Donner un exemple.

### Problème 3 : additionneur logique (20 points)

Codée sur un bit, la table de vérité de l'addition est donnée ci-dessous (**A** et **B** sont les entrées ; **S** est le résultat de la somme de **A** et **B**, et **C<sub>out</sub>** est la retenue) :

| A | B | S | C <sub>out</sub> |
|---|---|---|------------------|
| 0 | 0 | 0 | 0                |
| 0 | 1 | 1 | 0                |
| 1 | 0 | 1 | 0                |
| 1 | 1 | 0 | 1                |

a) Dessiner le circuit logique correspondant.

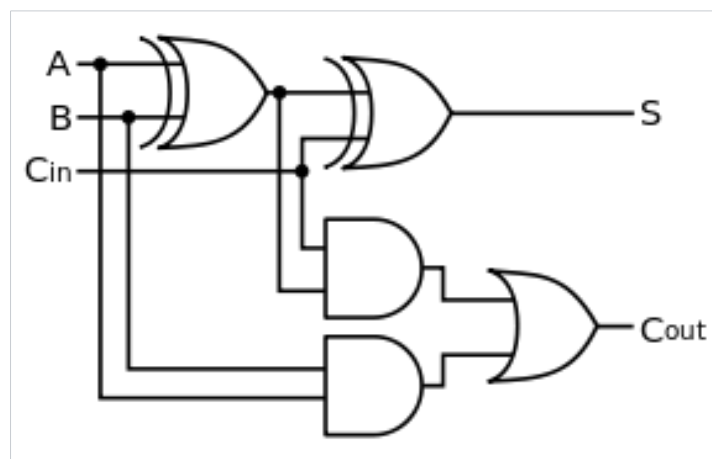
Un additionneur *complet* nécessite une entrée supplémentaire : une **retenue**, notée **C<sub>in</sub>**. L'intérêt de celle-ci est de permettre le chaînage des circuits.

b) Compléter la table de vérité ci-dessous.

| A | B | C <sub>in</sub> | S | C <sub>out</sub> |
|---|---|-----------------|---|------------------|
| 0 | 0 | 0               |   |                  |
| 0 | 0 | 1               |   |                  |
| 0 | 1 | 0               |   |                  |
| 0 | 1 | 1               |   |                  |
| 1 | 0 | 0               |   |                  |
| 1 | 0 | 1               |   |                  |
| 1 | 1 | 0               |   |                  |
| 1 | 1 | 1               |   |                  |

c) Donner les fonctions logiques pour **S** et **C<sub>out</sub>**.

d) Comparer la table de vérité correspondant au circuit ci-dessous avec celle du point b). Que peut-on en conclure ?



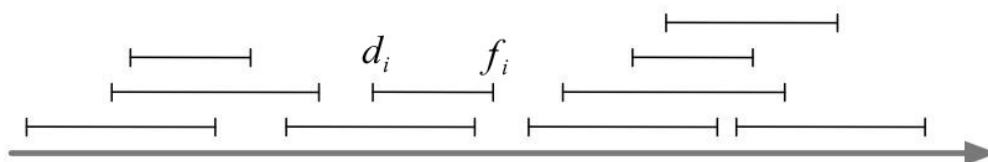
## Problème 4 : Python 3 (20 points)

```
n = 1001
liste = [1]*n
for i in range(2,int(sqrt(n))+1):
    if liste[i]==1:
        for j in range(i*2, n, i):
            liste[j] = 0
for i in range(2,n):
    if liste[i]==1:
        print(i, end=' ')
```

- Que fait la deuxième ligne de ce programme ? Pourrait-on la supprimer ? Pourquoi ?
- Ce programme produit l'erreur suivante : **NameError: name 'sqrt' is not defined**  
Quelle est la ligne qui pose problème ? Comment corriger ce problème ?
- Une fois corrigé, qu'écrira ce programme ? Quel est le nom de cet algorithme ?

## Problème 5 : Algorithmique (25 points)

Dans un cinéma, chaque séance  $i$  est caractérisée par l'intervalle  $(d_i, f_i)$ , où  $d_i$  est l'heure de début et  $f_i$  l'heure de fin. On peut représenter ces séances sur un schéma d'intervalles :



*Exemple de planning des séances de cinéma*

Vous voulez assister au maximum de séances dans une journée.

Vous considérez trois critères pour classer les séances, de la plus petite valeur à la plus grande :

- critère  $A$  : l'heure de début de la séance ( $d_i$ )
- critère  $B$  : la durée de la séance ( $f_i - d_i$ )
- critère  $C$  : l'heure de fin de la séance ( $f_i$ )

- Décrire en pseudo-code un algorithme glouton permettant de choisir les séances, après les avoir classées selon l'un des critères ci-dessus.
- Pour chacun des trois critères de classement, donner un exemple (s'il en existe) où votre algorithme ne donnera pas un choix optimal. Donner un exemple avec 5 séances sous forme d'un schéma d'intervalles comme celui de l'énoncé du problème.
- Appliquer votre algorithme aux séances ci-dessous. Donner le résultat pour chacun des trois critères de classement.

| Séance ( $i$ )        | 1   | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    |
|-----------------------|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Début ( $d_i$ )       | 9h  | 9h15  | 10h   | 13h   | 15h   | 15h15 | 16h   | 17h30 | 18h   | 19h30 |
| Fin ( $f_i$ )         | 11h | 10h50 | 11h20 | 15h25 | 16h40 | 18h15 | 18h05 | 19h   | 20h10 | 22h   |
| Durée ( $f_i - d_i$ ) | 120 | 95    | 80    | 145   | 100   | 180   | 125   | 90    | 130   | 150   |

- Vous voulez absolument assister à la séance 9. Modifier votre algorithme pour intégrer cette contrainte supplémentaire, puis répondre à nouveau à la question c.