

Nom : .....

Prénom : .....



**Informatique et systèmes  
de communication ISC**



Haute Ecole d'Ingénierie  
Hochschule für Ingenieurwissenschaften

---

## TEST INTERMÉDIAIRE

### 101.1 – Programmation impérative

---

#### Consigne :

Lisez attentivement la donnée et répondez de manière **lisible** aux questions. Vous avez droit pour cet examen à un aide-mémoire de 1 pages (1 feuille recto). Aucun moyen électronique n'est permis.

Un conseil : ne restez pas bloqués sur une question. Répondez tout d'abord aux questions avec lesquelles vous êtes à l'aise et revenez ensuite aux questions posant problème. Le barème indiqué est indicatif.

Question	Points	Bonus	Score
Short questions	10	2	
Loops analysis	8	0	
EBNF grammars	6	0	
Code comprehension	3	0	
Writing functions	7	1	
String manipulations	11	0	
Too fast, too furious	5	0	
Total:	50	3	

This exam has 7 questions, for a total of 50 points.

Rev 1.04ω

## Question 1 – Short questions (10 points)

Cette question est séparée en plusieurs exercices indépendants. Le nombre de point pour chaque exercice est indiqué dans la marge.

[4 Pt] (a) Soient les déclarations suivantes :

```
1 val foo : Int = 0xBABA
2 val foobar : Float = 3.825f
3 val baz : String = "y"
```

Les expressions suivantes sont correctes. Donnez le **type** et la **valeur** des expressions suivantes.

1) (foobar + - foobar).toByte

1) \_\_\_\_\_

2) (foo & 0xFF0).toHexString

2) \_\_\_\_\_

3) (((foo >> 4) << 4) | 0xE).toHexString

3) \_\_\_\_\_

4) (foo^foo).toShort

4) \_\_\_\_\_

5) ((foobar \* 100).toInt / 100.0f)

5) \_\_\_\_\_

6) if(foo < 0xFFFF) 'a' else 'b'

6) \_\_\_\_\_

7) if(true) '1' + baz else baz + '1'

7) \_\_\_\_\_

8) baz + ('d' + 1.9).toChar + ('t'-1).toChar

8) \_\_\_\_\_

[1 Pt] (b) Quel est le contenu de r après l'exécution du code ci-dessous :

```
1 val s: String = "TikTok"
2 val r: String = StringUtils.charAt(s, StringUtils.length(s)-2) +
3 "" + StringUtils.charAt(s, 2)
```

.....

[1 Pt] (c) Soit le code suivant :

```
1 val a : Boolean = ...
2 val b : Boolean = a || !a
```

Quelle est la valeur de b? ☐ Elle dépend du contenu de a ☐ true ☐ false

[2 Bo] (d) Écrivez le code (sans fonction) permettant d'écrire tous les multiples de 79 plus grands que 1 et plus petits que 1000 sur la console.

.....  
 .....  
 .....  
 .....

[4 Pt] (e) Vrai ou faux?

.....		
(255+1).toByte == (0xFFFF+1).toShort	True <input type="checkbox"/>	False <input type="checkbox"/>
.....		
Un générateur de séquence peut générer des Double	True <input type="checkbox"/>	False <input type="checkbox"/>
.....		
0 to 10 est une séquence plus longue que 1 until 11	True <input type="checkbox"/>	False <input type="checkbox"/>
.....		
\ est le caractère d'échappement	True <input type="checkbox"/>	False <input type="checkbox"/>
.....		
Un programme qui compile fonctionne toujours correctement	True <input type="checkbox"/>	False <input type="checkbox"/>
.....		
Le compilateur détecte les erreurs de type	True <input type="checkbox"/>	False <input type="checkbox"/>
.....		
println() est une fonction qui retourne un String	True <input type="checkbox"/>	False <input type="checkbox"/>
.....		
Dans l'expression if (foo) a else b, a et b doivent être de même type	True <input type="checkbox"/>	False <input type="checkbox"/>
.....		

## Question 2 – Loops analysis (8 points)

Que vont afficher **exactement** les boucles suivantes sur la console?

(a)

```

1  var foo: Int = 3
2  var bar: Int = 8
3  var i : Int = 8
4
5  do {
6      bar -= bar / (i>>1)
7      foo += 1 & 0
8      println(bar + foo + 1)
9  } while (foo + 1 < bar)

```

Solution :

(b)

```

1  var j = 6
2  var i = 0
3  while (i != j) {
4      print(i + " " + j + " * ")
5      i += 1
6      j -= 1
7  }

```

Solution :

**Solution :**

(c)

```
1 var a: Int = 0xf0
2 var t: String = ""
3 while (a != 0) {
4     t = (if (a % 2 == 0) '0' else '1') + t
5     a /= 2
6 }
7 print(t)
```

Solution :

(d)

```
1 // println(5)
2 for (i: Int <- 3 to 7){
3     print(s"${i-1*2/3} ")
4 }
```

### Question 3 – EBNF grammars (6 points)

(a) Soit la grammaire suivante pour exp

```
factor ::= 'x' | 'y' | 'z' | parexpr
parexpr ::= '(' exp ')'
op ::= '+' | '-'
exp ::= (factor op exp) | factor
```

[1 Pt]

1) Donnez une production valide la plus courte possible pour exp.

.....

[1 Pt]

2) Donnez une production utilisant chacune des règles de la grammaire.

.....

[1 Pt]

3) Donnez une production de  $\text{exp}$  utilisant au moins deux fois la règle  $\text{parexpr}$ .

.....

[3 Pt]

(b) Écrivez la description EBNF de la grammaire *even-integer* qui reconnaît uniquement les entiers pairs. Par exemple, dans cette grammaire -6 et 34 sont valides alors que 3 et -23 ne le sont pas. On considère également que 0 et -0 sont valides.

[illegible]

**Question 4 – Code comprehension (3 points)**

Analysez la fonction suivante puis répondez aux questions ci-dessous.

```

1  def isL(x: Char): Boolean = {
2      if (x >= 'a' && x <= 'z') true else false
3  }
4
5  def isU(x: Char): Boolean = {
6      if (x >= 'A' && x <= 'Z') true else false
7  }
8
9  def le(x: Char): Boolean = {
10     isU(x) || isL(x)
11 }
12
13 def bar(word: String): Int = {
14     var p: Int = 0
15
16     for (i <- word) {
17         if (!le(i))
18             return p
19
20         p += 1
21     }
22     return -1
23 }

```

[2 Pt] (a) Expliquez avec des phrases à quoi sert la fonction bar ci-dessus.

.....

.....

.....

.....

[1 Pt] (b) Donnez **deux exemples** complets d'utilisation de bar permettant de démontrer votre explication.

.....

.....

.....

**Question 5 – Writing functions (7 points)**

[2 Pt] (a) Écrivez une fonction nommée foo qui prend un Double nommé  $x$  en argument et qui retourne la valeur

$$foo(x) = \frac{3x^2 + \sin(x) - 3}{x^3}$$

.....

.....

.....

.....

.....

.....

[illegible]

This image shows a full page of a document template designed for handwritten notes or answers. It features approximately 20 evenly spaced horizontal dotted lines across the entire width of the page, providing a guide for letter height and placement. The background is plain white, and there are no margins, headers, or footers visible.

.....



On considère par simplification que la chaîne en entrée est toujours en minuscule (donc pas besoin de gérer les lettres majuscules). Exemple :

Écrivez la fonction `noTriples` qui corrige cette erreur dans la chaîne passée en argument. Notez que cette fonction n'interagit pas avec la console. Exemple :

Page 8/12





**Question 7 – Too fast, too furious (5 points)**

Afin d'améliorer la sécurité sur les routes, les pandores valaisans vont installer un nouveau radar ultra moderne. Ce radar est en effet capable de calculer la vitesse moyenne entre deux points séparés par une certaine distance.

Une infraction est alors constatée si la vitesse moyenne du véhicule est supérieure à la limitation de vitesse en cours.

Écrivez les fonctions suivantes :

- `def averageSpeed(timeOne : Double, timeTwo : Double, distance : Double)` retournant la vitesse moyenne entre les deux radars. Les deux paramètres de temps sont en secondes (remarque : seule la différence entre les deux temps est importante) et la valeur de retour doit être en km/h. La distance entre les deux radars est donnée en mètres.
- `isFaster`, un prédicat prenant comme arguments la vitesse du chauffeur et la limitation de vitesse et qui retourne si le chauffeur est en dehors des limitations.
- `getFineAmount(driverSpeed : Double, maxSpeed : Int)` retournant la valeur de l'amende que le chauffeur devra payer en cas d'infraction.

Pour le dernier point, le montant de l'amende est calculé comme suit :

- une déduction de 5 % doit être faite sur la vitesse mesurée. Arrondissez le résultat de cette déduction à l'unité (ex. 3.4 donne 3 et 3.6 donne 4).
- la facture s'élèvera à 60.- par tranche entamée de 10 km/h supérieurs à la limitation. Ainsi, le montant de l'amende doit être arrondi à la dizaine supérieure (23 km/h en trop vont être facturés comme 30 km/h).

Votre solution :

---

*Fin|Ende*

---