

NOM:**Code permanent:**

IFT1015 – Programmation I**Examen Intra – Hiver 2017****Chargée de cours :** Alena Tsikhanovich**Durée :** 1 heure 55 minutes**Directives :**

- Seule documentation permise: deux feuilles recto-verso (format « letter », et sans loupe!) résumant le cours.
 - Tous les appareils électroniques (cellulaire etc.) sont interdits à l'exception d'une montre pour connaître l'heure.
 - Inscrivez votre nom et code permanent sur l'entête du questionnaire (première page seulement);
 - Répondez aux questions directement sur la feuille du questionnaire dans les zone délimitées à cet effet;
 - Ne détachez pas les feuilles du questionnaire;
 - Veillez à la clarté de vos réponses et justifiez-les;
 - Si une question vous semble ambiguë, faites en une interprétation, expliquez-la et continuez.
-

On suppose que les programmes seront exécutés dans un environnement où une fonction `print` existe pour l'affichage (tel que dans `CodeBoot`).

Remarque: il n'y a aucune erreur de syntaxe dans les extraits de programmes ci-dessous.

Note :

| | |
|--------------|--------------|
| Question 1 | / 10 |
| Question 2 | / 15 |
| Question 3 | / 10 |
| Question 4 | / 15 |
| Question 5 | / 15 |
| Question 6 | / 25 |
| Question 7 | / 10 |
| Total | / 100 |

NOM:

Code permanent:

Question 1 : Variables et expressions (10 pts). VRAI OU FAUX

Indiquez **V** (pour vrai) ou **F** (pour faux) dans la colonne de droite pour chacune des affirmations équivalentes de la colonne de gauche.

| Affirmation | VRAI OU FAUX |
|--|--------------|
| 1. Le nombre $42e3$ en JavaScript est égal à 42×2^3 . | F |
| 2. Les nombres en JavaScript sont encodés sur 64 bits. | V |
| 3. <code>temp-max</code> est un identificateur valide en JavaScript | F |
| 4. Le résultat d'exécution <code>"Bonjour IFT1015!".length</code> est 15. | F(16) |
| 5. En JavaScript on peut comparer des chaînes de la même façon que les nombres: <code>"Aard" < "Zoro"</code> | V |
| 6. La fonction <code>print(10+20)</code> va imprimer <code>"1020"</code> . | F(30) |
| 7. <code>var x = test==1 ? "" : "s";</code> si la valeur de <code>test</code> égale 5 <code>x</code> contiendra la valeur <code>"s"</code> . | V |
| 8. Expression <code>15*4-4+88/2;</code> donnera comme résultat la valeur 100. | V |
| 9. Expression <code>-"15"*4-4+88/2;</code> donnera comme résultat la valeur -20. | V |
| 10. Fonction <code>typeof(...)</code> sert à identifier le type de l'expression en parenthèses. | V |

Question 2 : Variables et expressions (15 pts)

Donnez la représentation binaire des nombres suivants :

2.1 (1 pt) $0x34AF$

$$0x34AF = 0011010010101111_2$$

2.2 (1 pt) 129_{10}

$$129_{10} = 128 + 1 = 2^7 + 2^0 = 10000001_2$$

NOM:

Code permanent:

Voici la représentation d'un nombre :

| | | | |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
|---|---|---|---|

2.3 (1 pt) Donnez la valeur décimale de ce nombre en considérant qu'il s'agit de la représentation complément à 2 sur 4 bits.

| | | | |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
|---|---|---|---|

$$2^3 + 2^2 + 2^1 + 0 = -2$$

2.4 (1 pt) Donnez la valeur décimale de ce nombre en considérant qu'il s'agit de la représentation non signée sur 4 bits.

| | | | |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
|---|---|---|---|

$$2^3 + 2^2 + 2^1 + 0 = 14$$

2.5 (1 pt) Donnez 4 valeurs spéciales de type number .

NAN
+infini
-infini
Zéro négatif

2.6 (10 pts) Indiquez, à côté de chaque déclaration de variable, la valeur que contiendra la variable et son type. Pour les types string, écrivez la valeur entre "".

| Déclaration de variable | Type | Valeur |
|------------------------------------|---------|--------|
| var n1 = 2 + 2; | number | 4 |
| var n2 = 10e2; | number | 1000 |
| var n3 = 10e2+3; | number | 1003 |
| var n4 = n1*n1; | number | 16 |
| var n5 = "Hello World!".charAt(0); | string | "H" |
| var n6 = "5"; | string | "5" |
| var n7 = +n6; | number | 5 |
| var n8 = n6+n7; | string | "55" |
| var n9 = "Aard" < "Zoro"; | boolean | true |
| var n10 = +("Aard" < "Zoro"); | number | 1 |

NOM:**Code permanent:**

Question 3 : Affectations (10 pts)

Pour chaque énoncé `print` dans la colonne gauche, écrivez ce qu'il affiche dans la colonne droite.

Supposez que les variables `n1` - `n3` sont déclarées comme suit :

```
var n2 = "7" == 7;  
var n1 = 4;  
var n3 = 2;
```

| Énoncé <code>print</code> | Résultat de l'affichage |
|--------------------------------------|-------------------------|
| <code>print(n2);</code> | true |
| <code>print("7"=== 7);</code> | false |
| <code>print(Math.max(1+3,4));</code> | 4 |
| <code>print(2*n2*n2);</code> | 2 |
| <code>print(n1 += 4);</code> | 8 |
| <code>print(n1--);</code> | 8 |
| <code>print(n3 << 1);</code> | 4 |
| <code>print(+!n2);</code> | 0 |
| <code>print(n1+n3+"7");</code> | "97" |
| <code>print(n3%3);</code> | 2 |

NOM:

Code permanent:

Question 4. Exécution conditionnelle (15 pts)

Qu'affiche le programme suivant (écrivez-le à droite des « // »). Il y a 5 messages à identifier.

```
var team, ville;
var ville = "Montreal";
switch(ville) {
    case "Toronto":
        team = "Maple Leafs";
        break;
    case "New York":
        team = "Rangers";
        break;
    case "Pittsburgh":
        team = "Penguins";
        break;
    case "Ottawa":
        team = "Senators";
        break;
    case "Calgary":
        team = "Flames";
        break;
    case "Chicago":
        team = "Blackhawks";
        break;
    default:
        team = "Canadiens";
};
var msg = ville + " " + team;
print(msg); // 1. Montreal Canadiens

if( team.charAt(0) == "C" && ville.charAt(0) == "M")
    print("Go Habs Go!");
else
    print("BOO!");
// 2. Go Habs Go!

var mot = msg.substring(9,14) + msg.substring(6,7);
print(mot); // 3. Canada

if(ville == "Ottawa")
    print("Ottawa Senators");
else
    print("Chicago Blackhawks");
// 4. Chicago Blackhawks

if(team == "Maple Leafs" || team == "Senators" || team == "Flames"
|| team == "Canadiens")
print("Canada");
else
print("USA");
// 5. Canada
```

NOM:

Code permanent:

Question 5. Boucles (15 pts)

Écrire une somme des carrés des nombres impairs de 1 à 1000 en utilisant les boucles suivantes :

a) (5 pts) **do-while**

```
var nbImpaire = 1;
var sommeCarreNbImpaires = 0;

do {
    sommeCarreNbImpaires += nbImpaire*nbImpaire;
    nbImpaire = nbImpaire + 2;
}while(nbImpaire < 1000);

print(sommeCarreNbImpaires);
```

b) (5 pts) **while**

```
nbImpaire = 1;
sommeCarreNbImpaires = 0;

while(nbImpaire < 1000){
    sommeCarreNbImpaires += nbImpaire*nbImpaire;
    nbImpaire = nbImpaire + 2;
}

print(sommeCarreNbImpaires);
```

c) (5 pts) **for**

```
nbImpaire = 1;
sommeCarreNbImpaires = 0;

for(var i = 1; i < 1000; i+=2)
    sommeCarreNbImpaires += i*i;

print(sommeCarreNbImpaires);
```

NOM:

Code permanent:

Question 6. Fonctions (25 pts)

- a) (10 pts) Écrivez une fonction qui simule un dé. Cette fonction doit générer un nombre aléatoire entre 1 et 6.

```
var de = function ()  
{  
    return Math.floor(6*Math.random())+1;  
};
```

- b) (10 pts) Écrivez une fonction qui simule un lancer de deux dés en retournant la somme des deux nombres aléatoires entre 1 et 6. Indice : faites 2 appels de votre fonction de la sous question (a).

```
var DeuxDes = function ()  
{  
    var somme = de() + de();  
    return somme;  
};
```

- c) (5 pts) Écrivez une fonction effectuant des tests unitaires de deux fonctions (a) et (b) (pour vérifier ses bons fonctionnements).

```
var testDeuxDes = function()  
{  
    assert(de() >= 1 && de() <= 6);  
    assert(deuxDes() >= 2 && deuxDes() <= 12);  
};
```

NOM:

Code permanent:

Question 7. Appels de fonctions, passage de paramètres (10 pts = 5 + 5)

Dans le programme suivant, écrivez à côté de chaque `print` ce qu'il affiche (écrivez-le à droite du `//`).

```
a) var n1 = 5;
   var res = function (n) { var n1 = n*3+2; print(n1); };

   res(10);    // 1. 32

   print(n1);  // 2. 5

b) var k = -6;
   var x = 0;

   var res = function (x,y) { x *= y; k += x; };

   res(2, 1);

   print("k=" + k); // 3. -4

   print("x=" + x); // 4. 0
```