

Jeudi 11 décembre 2014 – 15h30 à 18h30

Directives:

1. Vous avez le droit à 2 pages (US letter) de notes. Les calculatrices électroniques, ordinateurs, tablettes et autres appareils du genre sont **interdits**. Vous n'avez pas le droit d'utiliser internet. Vous pouvez vous servir de votre téléphone cellulaire pour avoir l'heure.
2. Inscrivez tout de suite votre **nom**, votre **code permanent** et le **numéro de votre place** (s'il existe).
3. Répondez **sur le questionnaire**, dans l'espace libre qui suit chaque question. Si vous manquez de place, Écrivez au verso en l'indiquant très clairement. N'hésitez pas à commenter vos codes (en cas d'erreurs, cela peut vous sauver des points).
4. Le barème est donné à titre indicatif seulement. Votre note dépendra entre autre de l'élégance et de la simplicité de vos solutions.

BONNE CHANCE!!

1.	_____	/14
2.	_____	/12
3.	_____	/15
4.	_____	/14
5.	_____	/12
6.	_____	/16
7.	_____	/7
Total	_____	/ 90

Nom: _____ Code permanent: _____

Numéro de votre place: _____

1. (14 points) — Échauffement

Dans cette question, si vous pensez qu'un code est incorrect et ne peut s'exécuter, indiquez le clairement et préciser la nature du problème. Répondez aux questions qui suivent en considérant les déclarations:

```
var a=1, b=3;
```

(a) (2 points) Que produit l'exécution de: `print(a+b+" "+b+a, "=", 2*(a+b))` ;

(b) (2 points) Que produit l'exécution de: `print(a/b)` ;

(c) (2 points) Que produit l'exécution de: `print(a-b+1-b*a)` ;

(d) (2 points) Que produit l'exécution de: `print((Math.random() == 1)? true:false)` ;

(e) (2 points) Que produit l'exécution de: `print([a, "b", b])` ;

(f) (4 points) Que produit l'exécution du code suivant:

```
var quiz = function(r,t) {  
    r = t;  
    t[1] = r[2];  
    return r;  
} ;  
  
var t = [1,2,3];  
var r = ["a","b","c"];  
var s = quiz(t,r);  
print(t);  
print(r);  
  
s[1] = 5;  
print("t=",t,"r=",r,"s=",s);
```

2. (12 points) — Boucles simples

(a) (6 points) Écrire une fonction `loop` qui tire aléatoirement des nombres entiers entre 0 et 100 (inclus). Les tirages s'arrêtent lorsque la valeur 0 est tirée et la fonction affiche selon l'exemple donné ci-après:

- le nombre de tirages non nuls réalisés,
- le nombre d'entiers pairs tirés (la valeur 0 n'est pas comptée) et le pourcentage que cela représente par rapport au nombre total de tirages non nuls,
- la moyenne des entiers pairs tirés (la valeur 0 n'est pas comptée),
- la moyenne des entiers impairs.

Pour les tirages successifs: 1 5 3 69 56 0, l'affichage suivant doit être produit:

5 tirages >0, nbpairs: 1 (20.00%) moy pairs: 56.00 moy impairs: 26.80

- (b) (6 points) Écrire une fonction `loco` qui prend en paramètre un tableau d'entiers et qui affiche **tous** les éléments non (strictement) négatifs de ce tableau de telle sorte que le sens de lecture du tableau alterne lorsque qu'une valeur (strictement) négative est rencontrée. On commencera à afficher les valeurs de la gauche vers la droite, puis au premier entier négatif rencontré, de la droite vers la gauche, puis au prochain entier négatif rencontré de la gauche vers la droite, etc. Voici des exemples d'appels et de résultats:

```
loco([4,3,-1,3,-1,2,2,0,5,1])  produit 4 3 1 5 0 2 2 3
loco([3,2,-1,0,0,1,0,2,5,2])  produit 3 2 2 5 2 0 1 0 0
loco([2,2,-1,1,4,0,4,0,3,4])  produit 2 2 4 3 0 4 0 4 1
loco([4,4,4,3,3,-1,5,-1,5,3]) produit 4 4 4 3 3 3 5 5
loco([-1,1,0,2,2,0,-1,-1,2,3]) produit 3 2 1 0 2 2 0
```

3. (15 points) — Simplifiez-vous la vie

(a) (2 points) Expliquez clairement ce que retourne la fonction suivante:

```
var one = function(t) {  
  for (var i=0; i<t.length; ++i)  
    if (t[i] > 5) return true;  
  return false;  
};
```

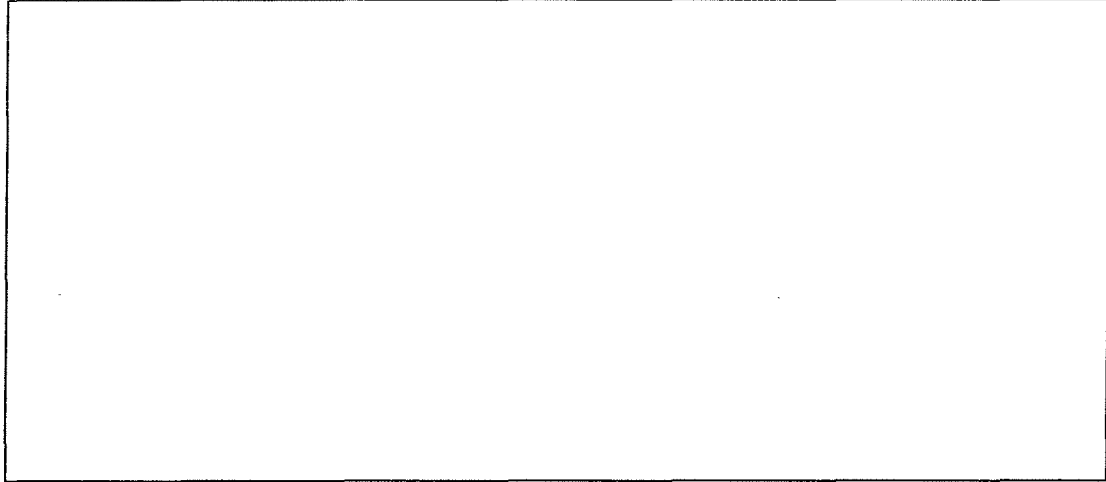
(b) (2 points) Simplifiez la fonction `one` sans en modifier la sémantique (c'est-à-dire que votre fonction doit retourner le même résultat pour la même valeur de paramètre) en utilisant un seul `return` et sans ajouter de variable (vous n'avez besoin que de `i`).

(c) (2 points) Quel est le résultat de l'exécution du code suivant?

```
var i = 6 + Math.random();  
print((i=5)? "cinq" : "autre");
```

(d) (2 points) Écrivez de la façon la plus simple possible une fonction équivalente à la fonction `two`:

```
var two = function() {  
    if (Math.random() < 1)  
        return true;  
    return false;  
};
```

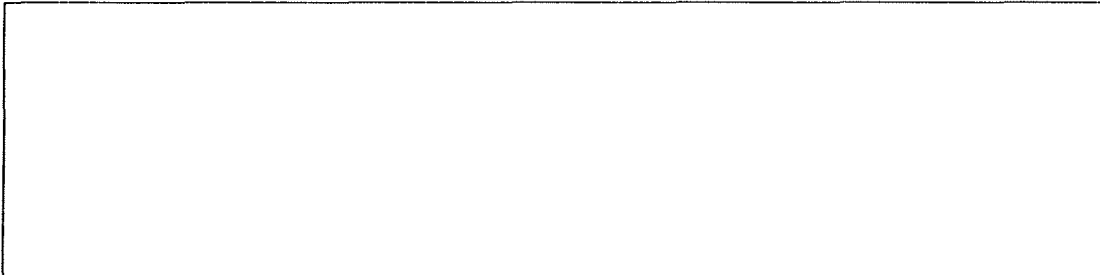


Considérez les deux fonctions suivantes:

```
var three = function(v) {  
    var sum = 0;  
    for (var i=0; i<10; ++i)  
        if (i !== v)  
            sum += i;  
  
    return sum;  
};
```

```
var three_c = function(v) {  
    var sum = 0;  
    for (var i=1; i<10; ++i) sum += i;  
    if ((v>=0) && (v<10)) sum -= v;  
    return sum;  
};
```

(e) (2 points) Que retourne `three(4)` ?



- (f) (2 points) Les fonctions `three` et `three_c` sont-elles équivalentes sémantiquement, c'est-à-dire retournent-elles le même résultat pour une même entrée (et ce, quelque soit l'entrée)? Justifiez.

- (g) (3 points) Considérez les deux fonctions suivantes:

```
var four = function(v,a,b) {          var four_c = function(v,a,b) {
    if ((v >a) && (v<b))                if ((v >a)) {
        return a;                      if (v<b)
    return b;                          return a;
};                                     }
                                     return b;
                                     };
```

Ces deux fonctions sont-elles équivalentes sémantiquement? Justifiez.

4. (14 points) — Récursivité

Considérez le code:

```
var quiz = function(a,b,r) {  
    if (a <= b)  
        return quiz(a+1,b,r+1) + quiz(a,b-1,r+1);  
    return r;  
};
```

(a) (2 points) La fonction quiz est-elle récursive terminale ? Justifiez.

(b) (2 points) Que retourne l'exécution de: `quiz(3,3)` ?

(c) (3 points) Que retourne l'exécution de: `quiz(3,4)` ?

(d) (2 points) Écrivez un code qui permet de saturer (remplir) la pile de récursivité.

- (e) (5 points) Écrire une fonction récursive `isPal(s)` qui prend une chaîne `s` en paramètre et qui retourne `true` si la chaîne constituée des seuls chiffres de `s` est un palindrome et `false` sinon. Si une chaîne ne contient aucun chiffre, `isPal` doit retourner `true`. Votre fonction ne doit créer aucun objet (pas de string, pas de tableau, etc.). L'appel `isPal("il a 25 ans et possède 5+2 chemises")` retourne `true` car 2552 est un palindrome. On admettra qu'une chaîne ne contenant aucun chiffre est un palindrome.

5. (12 points) — Algorithmique

- (a) (6 points) Écrire une fonction `longest` qui prend en paramètre un tableau d'éléments numériques et indique la séquence la plus grande de valeurs numériques triées en ordre croissant. Plus précisément, votre fonction doit retourner l'indice du début de la séquence et sa longueur. Voici des exemples d'appels de cette fonction où la séquence identifiée est marquée en gras:

```
print(longest([1, 2, 1, 4, 5, 2, 7, 8, 9])) produit 5, 4
```

```
print(longest([1, 2, 3, 4, 5, 6, 7, 2, 1, 2, 3, 4, 5, 6, 7, 8, 9, 3])) produit 8, 9
```

```
print(longest([5, 4, 3, 2, 1])) produit 0, 1
```

- (b) (6 points) Écrire une fonction `triangle(n)` qui prend un entier `n` (strictement positif) en paramètre et qui affiche un triangle de `n` lignes, dont les appels suivants illustrent l'aspect:

`triangle(2);`

`**`

`*`

`triangle(3);`

`***`

`**`

`*`

`triangle(4);`

`****`

`***`

`**`

`*`

6. (16 points) — Listes et arbres.

- (a) (2 points) Écrire du code qui permet de créer une *liste simplement chaînée* `l` qui contient trois maillons contenant respectivement les éléments 1, 2 et 4. Vous pouvez utiliser l'objet `Maillon` vu en cours ou bien créer le votre, mais vous ne pouvez pas utiliser l'objet `Liste` vu en cours (ou un équivalent).

- (b) (2 points) Écrire du code afin d'ajouter dans `l` le maillon d'information 3 entre le deuxième maillon (celui d'information 2) et le troisième (celui d'information 4). `l` doit donc contenir dans cet ordre les valeurs 1, 2, 3 et 4.

- (c) (3 points) À l'aide du constructeur `Node` vu en cours, créez un *arbre binaire de recherche* qui contient les valeurs: 5, 3, 1, 2, 7 et 8 et dont la racine contient la valeur 5. Dessinez cet arbre.

7. (7 points) — POO

Vous allez dans ce problème doter javascript des objets `De`, `Joueur` et `Jeu` qui permettent à des joueurs de jouer à un jeu de dés simple où chaque joueur tire chacun à son tour 2 dés et marque d points en cas de tirage d'un double où d est la valeur d'un dé. Le premier joueur arrivé au nombre de points exigé remporte la partie. Si au bout d'un nombre de tours spécifié, aucun joueur ne totalise le nombre de points exigés, alors la partie s'arrête sur un match nul. Lisez attentivement la totalité de ce problème puis répondez aux questions qui vous sont posées. Prenez note que la clarté des commentaires sera pris en compte lors de la notation.

- (a) (5 points) Écrivez un objet `De` qui permet de simuler un dé dont le nombre de faces (par défaut 6) est spécifié à la construction. Vous commenterez la signature du constructeur, ainsi que celle de toutes les méthodes que vous jugerez utiles à la résolution du problème. Voici un exemple de code qui doit fonctionner avec votre objet:

```
var d1 = new De(); // un de a 6 faces
var d2 = new De(8); // un de a 8 faces
print(d1,d2);
% 4 3
```

- (b) (5 points) Écrivez un objet `Joueur` qui permet de représenter un joueur par (au moins) son nom et son score. Là encore, vous devez commenter le constructeur ainsi que les méthodes utiles à la résolution du problème.

- (c) (6 points) Écrivez un objet `Jeu` qui prend en paramètre un tableau de noms de joueurs, puis de manière optionnelle le nombre de faces d'un dé (défaut 6), ainsi que le nombre de points à marquer pour gagner une partie (défaut: 21). Votre objet doit contenir (au moins) la méthode `play(verbose)` qui simule une partie où chaque joueur commence avec 0 point et tire les dés à tir de rôle. Si `verbose` vaut `true`, alors les détails de la partie seront affichés selon les spécifications implicites dans l'exemple qui suit (où la première ligne de trace indique qu'au premier tour, jean qui a un score de 0 fait un double 2 et gagne ainsi 2 points; la dernière ligne indique qu'au tour 8, jean qui a alors 16 points, fait un double 6 et remporte donc la partie, son score étant de 22, ce qui est supérieur ou égal à 21).

```
// des a 6 faces, partie en 21 points:
var game = new Jeu(["jean","paul","christine","jeanne"]);
game.play(true); // affiche les coups joues
%tour:1          jean score:0                (2 ,2) --> 2
%tour:1          paul score:0                (5 ,3)
%tour:1          christine score:0           (3 ,5)
%tour:1          jeanne score:0              (2 ,4)
%tour:2          jean score:2                 (3 ,3) --> 3
%tour:2          paul score:0                (6 ,2)
%tour:2          christine score:0           (2 ,5)
%...
%tour:7          jean score:10               (6 ,6) --> 6
%tour:7          paul score:0                (1 ,3)
%tour:7          christine score:0           (4 ,6)
%tour:7          jeanne score:4              (2 ,3)
%tour:8          jean score:16               (6 ,6) --> 6
%----- gagne !!! -----
```

Votre objet doit de plus produire un affichage identique (aux tirages aléatoires prêts) à celui prescrit par cette trace:

```
var game = new Jeu(["jean","paul","christine","jeanne"]);
print(game); // la partie n'est pas encore jouée
print("---");
print("je joue");
game.play();
print("---");
print(game); // la partie a tee jouee
%partie en 21 points avec 2 des a 6 faces
%opposant: jean,paul,christine,jeanne
%---
%je joue
%---
%partie en 21 points avec 2 des a 6 faces
%jouee en 25 coups
%Gagnant: paul score:23
%jean score:16 paul score:23 christine score:11 jeanne score:15
```