



Seule une feuille A4 recto-verso manuscrite est autorisée comme document.

Ce contrôle contient un questionnaire à choix multiples qui sera corrigé automatiquement. Il est donc important de remplir avec soin les réponses au **stylo noir**. Lorsque vous choisissez une réponse, il faut noircir complètement la case correspondante. **Ne pas toucher aux cases situées tout en haut de la feuille.**

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les questions ◇ sont ouvertes : il faut écrire la réponse dans le cadre et **ne pas noircir les cases de ces questions**. Les autres questions ont une unique bonne réponse.

Remplissez le cartouche de la copie-double anonymisée avec nom, prénom, numéro d'étudiant-e et signature puis coller le rabat du cartouche. **Reportez de façon très visible votre numéro de sujet sur la copie-double.** Le numéro de sujet est le premier nombre indiqué dans l'en-tête : par exemple il s'agit du sujet numéro 78 quand on lit +78/8/37+. **Reportez votre numéro d'anonymat présent sur la copie-double sur le sujet dans la grille ci-dessous.** Si votre numéro ne comporte que 6 chiffres, mettez un zéro (0) dans la première colonne.

<input type="checkbox"/>	0																
<input type="checkbox"/>	1																
<input type="checkbox"/>	2																
<input type="checkbox"/>	3																
<input type="checkbox"/>	4																
<input type="checkbox"/>	5																
<input type="checkbox"/>	6																
<input type="checkbox"/>	7																
<input type="checkbox"/>	8																
<input type="checkbox"/>	9																

1 Des promesses en javascript (/20)

On considère le code ci-dessous que l'on charge dans le navigateur.

```
1 function apply_delay(func, arg, delay = 1000, error = undefined){
2   return new Promise((resolve, reject) => {
3     setTimeout(() => {
4       if (error === undefined) {
5         resolve(func(arg))
6       } else {
7         reject(error)
8       }
9     }, delay);
10  });
11 }
12 function b(x) {
13   return x + "B";
14 }
```



Question 1 ♣ (/2). Soit la fonction javascript fa définie ci-dessous :

```
1 function fa() {
2   apply_delay(b, "A", 500, undefined)
3     .then(console.log)
4     .catch(console.error);
5   apply_delay(b, "B", 1000, undefined)
6     .then(console.log)
7     .catch(console.error);
8 }
```

On exécute `fa()`; au temps t_0 , indiquer ce qui s'affiche dans la console :

- "BB" à $t_0 + 1500$ ms
- "BB" à $t_0 + 1000$ ms
- "AB" à $t_0 + 1500$ ms
- "AB" à $t_0 + 500$ ms
- Aucune de ces réponses n'est correcte.

Question 2 ♣ (/2). Soit la fonction javascript fb définie ci-dessous :

```
1 function fb() {
2   apply_delay(b, "A", 500, undefined)
3     .then(console.log)
4     .then(() => apply_delay(b, "B", 1000, undefined))
5     .then(console.log)
6     .catch(console.error);
7 }
```

On exécute `fb()`; au temps t_0 , indiquer ce qui s'affiche dans la console :

- "AB" à $t_0 + 500$ ms
- "AB" à $t_0 + 1500$ ms
- "BB" à $t_0 + 1000$ ms
- "BB" à $t_0 + 1500$ ms
- Aucune de ces réponses n'est correcte.

Question 3 ♣ (/2). Soit la fonction javascript fc définie ci-dessous :

```
1 function fc() {
2   apply_delay(b, "A", 500, "Une erreur")
3     .then(console.log)
4     .catch(console.error);
5   apply_delay(b, "B", 1000, undefined)
6     .then(console.log)
7     .catch(console.error);
8 }
```

On exécute `fc()`; au temps t_0 , indiquer ce qui s'affiche dans la console :

- | | |
|---|---|
| <input type="checkbox"/> "AB" à $t_0 + 500$ ms | <input type="checkbox"/> "BB" à $t_0 + 1500$ ms |
| <input type="checkbox"/> "AB" à $t_0 + 1000$ ms | <input type="checkbox"/> "BB" à $t_0 + 1000$ ms |
| <input type="checkbox"/> "Une erreur" à $t_0 + 1000$ ms | <input type="checkbox"/> "BB" à $t_0 + 500$ ms |
| <input type="checkbox"/> "Une erreur" à $t_0 + 1500$ ms | <input type="checkbox"/> "AB" à $t_0 + 1500$ ms |
| <input type="checkbox"/> "Une erreur" à $t_0 + 500$ ms | <input type="checkbox"/> Aucune de ces réponses n'est correcte. |



Question 4 ♣ (/2). Soit la fonction javascript fd définie ci-dessous :

```
1 function fd() {
2   apply_delay(b, "A", 500, "Une erreur")
3     .then(console.log)
4     .catch(() => apply_delay(b, "B", 1000, undefined))
5     .then(console.log)
6     .catch(console.error);
7 }
```

On exécute fd(); au temps t_0 , indiquer ce qui s'affiche dans la console :

- | | |
|---|---|
| <input type="checkbox"/> "Une erreur" à $t_0 + 1000$ ms | <input type="checkbox"/> "BB" à $t_0 + 500$ ms |
| <input type="checkbox"/> "AB" à $t_0 + 500$ ms | <input type="checkbox"/> "AB" à $t_0 + 1500$ ms |
| <input type="checkbox"/> "AB" à $t_0 + 1000$ ms | <input type="checkbox"/> "Une erreur" à $t_0 + 1500$ ms |
| <input type="checkbox"/> "BB" à $t_0 + 1000$ ms | <input type="checkbox"/> "BB" à $t_0 + 1500$ ms |
| <input type="checkbox"/> "Une erreur" à $t_0 + 500$ ms | <input type="checkbox"/> Aucune de ces réponses n'est correcte. |

Question 5 ♣ (/2). Soit la fonction javascript fe définie ci-dessous :

```
1 function fe() {
2   apply_delay(b, "A", 500, "Une erreur")
3     .then(console.log)
4     .then(() => apply_delay(b, "B", 1000, undefined))
5     .then(console.log)
6     .catch(console.error);
7 }
```

On exécute fe(); au temps t_0 , indiquer ce qui s'affiche dans la console :

- | | |
|---|---|
| <input type="checkbox"/> "BB" à $t_0 + 500$ ms | <input type="checkbox"/> "AB" à $t_0 + 500$ ms |
| <input type="checkbox"/> "BB" à $t_0 + 1000$ ms | <input type="checkbox"/> "AB" à $t_0 + 1000$ ms |
| <input type="checkbox"/> "Une erreur" à $t_0 + 1000$ ms | <input type="checkbox"/> "Une erreur" à $t_0 + 1500$ ms |
| <input type="checkbox"/> "AB" à $t_0 + 1500$ ms | <input type="checkbox"/> "Une erreur" à $t_0 + 500$ ms |
| <input type="checkbox"/> "BB" à $t_0 + 1500$ ms | <input type="checkbox"/> Aucune de ces réponses n'est correcte. |

Question 6 ♣ (/2). Soit la fonction javascript va définie ci-dessous :

```
1 function va() {
2   let val = "Z";
3   apply_delay(b, "A", 500, undefined)
4     .then(res => {val = res; return val;})
5     .then(console.log)
6     .catch(console.error);
7   console.log(val);
8 }
```

On exécute va(); au temps t_0 , indiquer ce qui s'affiche dans la console :

- | | |
|--|---|
| <input type="checkbox"/> "Z" à $t_0 + 0$ ms | <input type="checkbox"/> "AB" à $t_0 + 0$ ms |
| <input type="checkbox"/> "AB" à $t_0 + 500$ ms | <input type="checkbox"/> Aucune de ces réponses n'est correcte. |
| <input type="checkbox"/> "Z" à $t_0 + 500$ ms | |



Question 7 ♣ (/2). Soit la fonction javascript vb définie ci-dessous :

```
1 function vb() {
2   let val = "Z";
3   apply_delay(b, "A", 500, undefined)
4     .then(res => res+"C")
5     .then(console.log)
6     .catch(console.error);
7   console.log(val);
8 }
```

On exécute `vb()`; au temps t_0 , indiquer ce qui s'affiche dans la console :

- | | |
|---|---|
| <input type="checkbox"/> "AB" à $t_0 + 0$ ms | <input type="checkbox"/> "ABC" à $t_0 + 500$ ms |
| <input type="checkbox"/> "ABC" à $t_0 + 0$ ms | <input type="checkbox"/> "AB" à $t_0 + 500$ ms |
| <input type="checkbox"/> "Z" à $t_0 + 500$ ms | <input type="checkbox"/> Aucune de ces réponses n'est correcte. |
| <input type="checkbox"/> "Z" à $t_0 + 0$ ms | |

Question 8 ◇ (/6) Dans le style le plus fonctionnel possible (c'est-à-dire sans affectation sur les paramètres des fonctions et en ne déclarant que des constantes), écrire une fonction qui prend en paramètre une `url` supposée retourner un tableau de mots (de type `string`) au format `json`. La fonction doit accéder à l'`url` et ajouter au paragraphe `<p id="parag"></p>` la concaténation des mots de la liste séparés par des espaces. En cas d'erreur, il faudra insérer un message dans l'élément `<div id="err"></div>`.

f i pj j



2 API fonctionnelle du prototype Array (/20)

La méthode `Array.prototype.every(p)` d'un tableau javascript teste si *tous* les éléments du tableau vérifie la propriété `p` passée en paramètre, par exemple `[1,2,3,4,5].every(x=>x%2 === 1)`; produit `false` tandis que `[1,3,5].every(x=>x%2 === 1)`; produit `true`. Similairement la méthode `Array.prototype.some(p)` teste si *au moins un* des éléments du tableau vérifie la propriété `p`.

Question 9 \diamond (/2) Définir une fonction `pourtous1` qui fait la même chose que `Array.prototype.every` en utilisant *uniquement* `filter`. f j

Question 10 \diamond (/2) Définir une fonction `pourtous2` qui fait la même chose que `Array.prototype.every` en utilisant *uniquement* `reduce`. f j

Question 11 \diamond (/2) Définir une fonction `filtre` qui fait la même chose que `Array.prototype.filter(p)` mais en utilisant *uniquement* `reduce`. f j

Question 12 (/1). Soit le tableau `tab = [1,2,3,4,5,6,7,8,9]`, donner le résultat de l'évaluation de `tab.some(n => n % 3 === 0 && n % 5 === 0)`;

- true false
 Cet appel produit une erreur à l'exécution.

Question 13 (/1). Soit le tableau `tab = [1,2,3,4,5,6,7,8,9]`, donner le résultat de l'évaluation de `tab.some(n => n % 3 === 0) && tab.some(n => n % 5 === 0)`;

- false Cet appel produit une erreur à l'exécution.
 true

Question 14 (/1). Soit le tableau `tab = [1,2,3,4,5,6,7,8,9]`, donner le résultat de l'évaluation de `tab.filter(n => n % 2 === 0).map(n => 1 + n)`;

- Cet appel produit une erreur à l'exécution. [2, 4, 6, 8, 10]
 [2, 3, 4, 5, 6, 7, 8, 9, 10] [3, 5, 7, 9]



Question 15 (/1). Soit le tableau `tab = [1,2,3,4,5,6,7,8,9]`, donner le résultat de l'évaluation de `tab.filter(n => n % 2 === 0).map(n => n*n).reduce((acc,n2)=> acc+n2, 0);`

 42
 20 120
 Cet appel produit une erreur à l'exécution.

Question 16 ◇ (/2) Que calcule la fonction `let fred = (arr)=> arr.reduce((acc,x)=> acc || x, true);` f j

Question 17 ◇ (/2) Que calcule la fonction `let fred = (arr)=> arr.reduce((acc,x)=> acc || x, false);` f j

Question 18 ◇ (/2) Que calcule `let fred = (arr)=> arr.reduce((acc,x)=> acc + 1, 0);` f j

Question 19 ◇ (/4) Écrire, le plus fonctionnellement possible, une fonction qui prend en paramètre en tableau de chaînes et renvoie la tableau des jours de la semaine des chaînes qui sont des dates valides. Par exemple, pour `['2019-05-22', 'examen', {}, '2019-05-23']` on doit obtenir `["mercredi", "jeudi"]`. Notez que pour un objet `ma_date` de type `Date`, la méthode `ma_date.toLocaleDateString('fr', {weekday : 'long'})` donne son jour de la semaine et que `let check = d => d instanceof Date && !isNaN(d);` teste si un objet `Date` est valide. f i pj j



3 Problème de modélisation : les paires de Church (/20)

Dans le TD1, une représentation (ou codage) des booléens dans le λ -calcul avait été étudiée : après avoir codé « vrai » et « faux » par des termes du λ -calcul, on avait proposé des termes représentant les opérations « et », « ou » etc. On rappelle le terme qui représente « vrai » $\mathbf{TT} = \lambda x.\lambda y.x$.

Ce problème suit la même idée, mais il s'agit cette fois de représenter des paires et leurs opérations. La paire (ordonnée) notée $\langle M, N \rangle$ constituée de deux éléments M et N est représentée en λ -calcul par le terme $\lambda p.((p M) N)$. En convenant de l'associativité à gauche de l'application, on peut omettre les parenthèses et simplement écrire $\lambda p.p M N$. On définit également la première projection $\pi_1 = \lambda P.P(\mathbf{TT})$. Cette représentation est appelée *paires de Church*.

Question 20 \diamond (/4) Soit la paire $\langle M, N \rangle$ et P sa représentation en λ -calcul. Prouver que la définition de la première projection est correcte, c'est-à-dire que $\pi_1 P \rightsquigarrow M$ f i pj j

Question 21 \diamond (/2) Donner une définition de la seconde projection π_2 , c'est-à-dire la fonction qui à P fait correspondre N , en supposant que P est la représentation en λ -calcul de $\langle M, N \rangle$. f j

Question 22 \diamond (/2) En utilisant les termes précédents, définir la fonction **swap** qui intervertit les éléments de la paire, c'est-à-dire qui à $\langle M, N \rangle$ fait correspondre $\langle N, M \rangle$. f j



Question 23 \diamond (/2) On appelle constructeur de la paire de Church le terme $\lambda M.\lambda N.\lambda p.p M N$. Donner son type. f j

Question 24 \diamond (/4) Prouver que le terme $\mathbf{assoc} = \lambda P.\lambda p.(\lambda q.(\pi_1 P)(\pi_1 (p_2 P)))(\pi_2 (p_2 P))$ exprime l'associativité des paires. Pour cela, évaluez \mathbf{assoc} sur $\langle X, \langle Y, Z \rangle \rangle$ f i pj j

Question 25 \diamond (/2) Définir en javascript une fonction `toChurch` qui prend en entrée deux arguments et renvoie leur paire de Church. f j

Question 26 \diamond (/2) Définir en javascript la première projection π_1 f j

Question 27 \diamond (/2) En javascript, en oubliant le λ -calcul, comment définiriez-vous une structure de paire ? f j