



### Contrôle Continu Final de LIFAP5 - Programmation fonctionnelle pour le Web 18/05/2021

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

Ne pas toucher aux cases situées tout en haut de la feuille, elles servent à repérer vos copies lors de la numérisation. Les questions sont ouvertes : il faut écrire la réponse dans le cadre et **ne pas noircir les cases de ces questions**. Il ne faut pas écrire en dehors des cadres, pensez à écrire la réponse au brouillon avant de la recopier.

Remplissez le cartouche de la copie-double anonymisée avec nom, prénom, numéro d'étudiant-e- et signature puis coller le rabat du cartouche. Reportez votre numéro d'anonymat présent sur la copie-double dans la grille ci-dessous. Si ce numéro ne comporte que 5 chiffres, mettez 0 comme premier chiffre.

<input type="text"/>						
<input type="text"/>						
<input type="text"/>						
<input type="text"/>						
<input type="text"/>						
<input type="text"/>						
<input type="text"/>						
<input type="text"/>						
<input type="text"/>						
<input type="text"/>						

## 1 Transformations de tableaux

On dans cette partie, considère un tableau Javascript, stocké dans une constante nommée `tab` et contenant des objets ayant les champs suivants : `jour`, `compte`, `theme` et `lien`.

On considère la fonction `groupBy` qui prend en argument un tableau `t` et une fonction `f` et qui renvoie un tableau contenant des objets ayant chacun un champ `key` et un champ `values` tel que :

- Les champs `values` sont des tableaux contenant des éléments de `t`
- Si `o` est dans `t` alors il est dans un des éléments du résultat pour lequel le champ `key` vaut `f(o)`.
- Pour un résultat donné, les valeurs des champs `key` sont unique (c'est-à-dire que pour une valeur `k`, il y a au plus un élément du résultat dont le champ `key` vaut `k`)

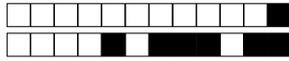
On a donc que si un objet `r` est dans `groupBy(t, f)`, alors `r.values` contient les éléments de `t` tels que `f(t)` a pour valeur `r.key`.

Par exemple, `groupBy([2,4,5,6,9], x => x % 2)` vaut `[{key: 0, values: [2,4,6]}, {key: 1, values: [5,9]}]`

On considère également la fonction `distinct` qui prend un tableau `t` et renvoie un tableau contenant les éléments de `t`, mais sans doublons.

On rappelle par ailleurs l'existence des méthodes `map`, `filter` et `reduce` pour les tableaux Javascript.

On rappelle également qu'une expression n'est pas une instruction et qu'en particulier, une expression ne contient pas de déclaration de variables.



**Question 1** (/1) Donner une expression Javascript permettant de récupérer un tableau contenant les éléments de `tab` dont le champ `compte` est strictement supérieur à 10.  f  j

**Question 2** (/1) Donner une expression Javascript permettant de récupérer un tableau contenant les valeurs du champ `compte` des éléments de `tab`.  f  j

**Question 3** (/2) Donner une expression Javascript permettant d'obtenir, pour chaque jour, la somme des champs `compte` des éléments de `tab` ayant ce jour comme valeur du champ `jour`. On utilisera la fonction `groupBy` dans cette expression.  f  i  pj  j

**Question 4** (/1) Donner une expression Javascript permettant d'obtenir, à partir d'un tableau `t` et d'une fonction `f`, le tableau des valeurs *distinctes* de `f(o)` pour `o ∈ t`.  f  j

**Question 5** (/1) Donner une expression Javascript permettant d'obtenir, à partir d'un tableau `t` d'une valeur `v` et d'une fonction `f`, le tableau contenant les objets `o` de `t` tels que la valeur de `f(o)` soit `v`.  f  j



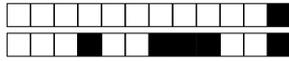
**Question 6** (/2) En utilisant les deux questions précédentes, donner une expression Javascript permettant de calculer `groupBy(t, f)`  f  i  pj  j

## 2 λ-calcul et Javascript

**Question 7** (/1) Soit  $M$  le terme de λ-calcul suivant :  $\lambda x. \lambda y. ((xy)y)$ . Donner le type de  $M$  en utilisant pour  $y$  un type générique  $\tau$ .  f  j

**Question 8** (/1) Traduire en Javascript le terme de λ-calcul suivant :  $\lambda x. \lambda y. ((xy)y)$ .  f  j

**Question 9** (/1) Évaluer le terme de λ-calcul suivant :  $(\lambda x. \lambda y. ((xy)y))(\lambda z. \lambda u. zu)(\lambda w. \lambda p. (w(wp)))$   f  j



**Question 10** (/1) On rappelle que l'instruction `const x = expr1; return expr2` de Javascript peut être traduite en  $\lambda$ -calcul par  $((\lambda x.M_2)M_1)$  si  $M_1$  est la traduction de `expr1` et  $M_2$  celle de `expr2`. Traduire en  $\lambda$ -calcul la fonction Javascript suivante :

```
1 function (x) {  
2   const v = (y) => x(y);  
3   return (z) => v(z);  
4 }
```

 f  j

**Question 11** (/1) On rappelle que l'instruction `const x = expr1; return expr2` de Javascript peut être traduite en  $\lambda$ -calcul par  $((\lambda x.M_2)M_1)$  si  $M_1$  est la traduction de `expr1` et  $M_2$  celle de `expr2`. Réécrire la fonction suivante sans utiliser `const` en la traduisant en  $\lambda$ -calcul puis en la retraduisant le résultat en Javascript.

```
1 function (x) {  
2   const a = (y) => y(x);  
3   return (z) => z(a);  
4 }
```

 f  j

### 3 Javascript et programmation Web

On considère que le serveur Web qui répond à l'URL `https://exemple.com/data` envoie des documents JSON ayant la forme donnée dans la figure 1. On considère également une page Web contenant entre autres le code HTML de la figure 2. On suppose enfin que cette page charge le script donné dans la figure 3.

```
1 [  
2   {  
3     "titre": "titre1",  
4     "description": "une tache a realiser",  
5     "echeance": "2021-05-18",  
6     "duree": 60  
7   },  
8   ...  
9 ]
```

FIGURE 1 – Exemple de données envoyées par le serveur



```
1 <input id="input-filtre" type="text" />
2 <table>
3   <thead>
4     <tr>
5       <th>Date</th>
6       <th>Titre</th>
7     </tr>
8   </thead>
9   <tbody id="liste-taches"></tbody>
10 </table>
```

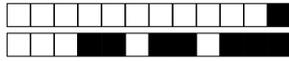
FIGURE 2 – Extrait du code HTML de la page

```
1 const loadData = () => fetch("https://exemple.com/data").then(response =>
   response.json()).then(afficheListe);
2
3 const afficheListe = (listeTaches) => {
4   // TODO
5 };
6
7 const changeFiltre = () => {
8   // TODO
9 };
10
11 document.addEventListener("DOMContentLoaded", function(){
12   loadData();
13   document.getElementById("input-filtre").onchange = changeFiltre;
14 });
```

FIGURE 3 – Code Javascript

**Question 12** (/2) Implémenter la fonction `afficheListe` qui devra insérer, pour chaque objet tâche du tableau `listeTaches` passé en paramètre, une ligne dans le corps (`tbody`) la table HTML de la figure 2. On rappelle qu'en HTML, une ligne correspond à un élément `<tr>` et une colonne à un élément `<td>`.

f  i  pj  j



**Question 13** (/2) L'attribut `onchange` d'un élément HTML contient une fonction appelée à chaque fois que la valeur de l'élément est modifiée. Implémenter la fonction `changeFiltre` ci-dessus de façon à ce que la table de tâches affiche uniquement les tâches dont le titre contient la valeur du champ `<input>` ci-dessus.

On pourra au besoin recharger les données depuis le serveur. On rappelle par ailleurs que la méthode `String.prototype.includes(aChercher)` permet de tester si une chaîne de caractères contient l'argument `aChercher`.

f  i  pj  j

**Question 14** (/2) On suppose maintenant que si on passe un paramètre `search` au serveur lors de la requête sur `https://exemple.com/data`, ce dernier ne renverra que les tâches dont le titre contient la valeur du paramètre. Par exemple l'URL `https://exemple.com/data?search=toto` ne renverra que les tâches dont le titre contient `toto`. Modifier le code de la fonction `changeFiltre` de façon à ce que le filtrage des tâches se fasse sur le serveur.

f  i  pj  j