Lycée Secondaire Hammam-Sousse	**Devoir de contrôle n°2** S.T.I - Pratique	**Matière :** S.T.I **Classe :** 3INFO **Durée :** 1h **Enseignant :** Mohamed Anis MANI **Année Scolaire :** 2022-2023

Nom & Prénom : .....

Un site marchand offre à ses client la possibilité de paiement en ligne de leurs achats. La page de paiement se présente comme suit :

Travail demandé

1. Créer un dossier portant votre nom & prénom sur le bureau, puis l'ouvrir à l'aide de Visual Studio Code.
2. Créer un fichier nommé **paiement.html** contenant le formulaire ci-dessus.
   - Le ***Numéro de la carte*** (champ **text**) indique le numéro de la carte bancaire du client : une séquence de 16 chiffres,
   - Les deux listes de choix (champ **select**) ***Valable jusqu'au*** sont utilisées pour choisir le mois (Janvier - Décembre) et l'année d'expiration (2023 – 2026) de la carte bancaire respectivement,
   - Le, Card Verification Value, ***CVV2*** (champ **password**) est la clé de validation de la carte qui se présente sous la forme d'un nombre de trois chiffres imprimés au dos de la carte bancaire,
   - Le ***Nom du détenteur*** (champ **text**) est le nom du propriétaire de la carte bancaire,
   - L'***Adresse e-mail*** (champ **email**) est l'adresse email de confirmation de paiement,

   Tous les champs sont obligatoires et ne doivent pas être laissés vides.
3. Créer un fichier nommé **paiement.css**. Placer la mise en forme du formulaire dans ce fichier.
4. Créer un fichier nommé **paiement.js**. Et y placer le code JavaScript nécessaire pour la validation de l'opération de paiement.
5. Le clic sur le bouton **Paiement** permet de vérifier :
   - que **tous les champs sont présents**, si un champ est vide le message : "Tous les champs sont obligatoires" est affiché,

- que le **numéro de carte** est valide, sinon le message "Numéro carte invalide" est affiché, développer une fonction **`validCardNumber(cardNum)`** basé sur l'algorithme de Luhn décrit ci-dessous,
- que le **CVV2**, le **nom du détenteur** et l'**email** sont valides, sinon un message d'erreur est affiché pour chaque champ,

6. La validation du numéro de la carte bancaire utilise l'**algorithme de Luhn**.

L'algorithme de Luhn détermine si un numéro de carte de crédit est valide ou non. Pour un numéro de carte de crédit donné :

a) Doublez la valeur des chiffres d'indices pairs, en commençant du gauche.
b) Ajoutez les chiffres du résultat de l'étape (a) aux chiffres restants du numéro de la carte de crédit.
c) Si le **résultat mod 10 est égal à 0**, le numéro est valide. S'il est différent de 0, la validation échoue.

Exemple

Validons le numéro de carte de crédit Visa portant le numéro 4624 7482 3324 9080.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
	4	6	2	4	7	4	8	2	3	3	2	4	9	0	8	0	
	x2		x2		x2		x2		x2		x2		x2		x2		
(a)	8	↓	4	↓	14	↓	16	↓	6	↓	4	↓	18	↓	16	↓	
(b)	8	+6	+4	+4	+1+4	+4	+1+6	+2	+6	+3	+4	+4	+1+8	+0	+1+6	+0	=73

(c) 73 mod 10 = 3 ⇒ **Carte invalide**

Validons le numéro de carte de crédit Visa portant le numéro 4624 7482 3324 9780.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
	4	6	2	4	7	4	8	2	3	3	2	4	9	7	8	0	
	4x2=8		2x2=4		7x2=14		8x2=16		3x2=6		2x2=4		9x2=18		8x2=16		
	8	+6	+4	+4	+1+4	+4	+1+6	+2	+6	+3	+4	+4	+1+8	+7	+1+6	+0	=80

→ 80 mod 10 = 0 ⇒ **Carte valide**

<u>Les fonctions suivantes peuvent être utilisées :</u>

```
//-- Validation d'une adresse email
function validMail(mail) {
  return /^[\w-\.]+@([\w-]+\.)+[\w-]{2,4}$/g.test(mail);
}
//-- Validation du CVV2
function validCVV2(CVV2) {
  return /^[0-9]{3}$/g.test(CVV2);
}
//-- Vérification de la validité du nom
function estAlphabetique(nom) {
  return /^[\w\s]{3,}$/g.test(nom);
}
//-- Générer un nombre aléatoire dans l'intervalle [a, b]
function randint(a, b) {
  return Math.floor(a + Math.random() * (b - a + 1));
}
```

HTML + CSS	JavaScript	Total
Form , Controls , Link , Script , CSS 0.5 + 3.5 + 0.5 + 0.5 + 3	Elts, btn ev., handler fn, verif, valid card fn, oth. Fn 7*.25+ .75+ .75+ 6*.75+ 2.75+ 3*.5	
/ 8	/ 12	