ISET BIZERTE

LA BIBLIOTHEQUE

DEPARTEMENT

TI

EXAMENS JUIN

1ERE

TI

ANNEE UNIVERSITAIRE

2020/2021

	MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE Direction Générale des Etudes Technologiques **Institut Supérieur des Etudes Technologiques de Bizerte** **Département Sciences économique de gestion**
Examen **Matière : Technique d'expression**	
Enseignante : Souli Zohra	Documents : Non autorisés
Date : 17/ 06 /2021	Durée : 1h30
Nombre de pages : 3	Classe: .....

Question 1 : Cocher la bonne réponse (**5pts**)

1) Choisissez la /les phrase(s) correcte(s) pour commencer votre lettre
   a) Votre entreprise, leader de son marche […]
   b) Lors du salon XXX, j'ai pris connaissance de vos opportunités de développement […]
   c) Disposant de compétences solides pour la poste XXX à pour voir […]
2) Quel type de CV est basé sur l'expérience ?
   a) Fonctionnel
   b) Chrono-fonctionnel
   c) Chronologique
3) Quels (s) éléments (s) faut-il mentionner dans l'objet ?
   a) L'intitulé du poste à pourvoir
   b) La référence de l'entreprise
   c) Le nom de l'entreprise
4) Quelle est la longueur idéale d'une lettre de motivation?
   a) Moins de 10 lignes
   b) Autour de 15 lignes
   c) Au-delà d'une page
5) Faut-il parler de son CV lors de la rédaction de la lettre?
   a) Oui, il faut le réciter
   b) Oui, mais il faut juste s'y appuyer pour convaincre
   c) Non, c'est un document à part
6) Ces 3 propositions sont souvent mal écrites dans des lettres de motivation .Retrouvez celle qui est écrite convenablement
   a) Je vous serai gré
   b) Votre chiffre d'affaire
   c) Veuillez agréer
7) Quelle phrase est correctement écrite ?
   a) A l'attention du responsable
   b) A l'intention du responsable

8) Trois des termes suivent sont des synonyme qui est l'intrus?
   a) Messagerie électronique
   b) Courriel
   c) E-mail
   d) Forum
9) Qu'est-ce que un e-mail?
   a) Un courrier électronique
   b) Une lettre à poster
   c) U synonyme d'Internet
   d) Je ne sais pas
10) Pour écrire un courrier électronique j'ai besoin :
   a) Du papier et une enveloppe
   b) De l'adresse électronique du destinateur
   c) Du navigateur de mon destinateur

Question2 :Dites si l'énoncé est vrai ou faux. (2.5pts)

	Vrai	Faux
a) Pour envoyer un message à une personne on doit avoir son mot de passe		
b) Il faut faire un double de notre CV		
c) Il faut adapter la lettre de motivation à la fonction pour laquelle vous postulez		
d) La mise en page d'une lettre de motivation doit être formelle		
e) Pour envoyer un message à une personne on doit avoir son adresse e-mail		

Question3 :Ya-t-il des règles ou des erreurs à éviter pour bien écrire une lettre de motivation. (2pts)

.....

.....

.....

.....

Question4: Définissez le terme « **email** » (**2**pts)

.....
.....
.....
.....

Question45 : Donnez le rôle de chaque zone de saisie :(**1.5**pts)

A :.....

Cc :.....

Objet :.....

Question6 : Rédigez un **email** où vous réclamez au service après –vente la défectuosité d'un PC portable. Veuillez respecter les différentes parties de l'email. (7pts)

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Département Technologies de l'Informatique (TI)

EXAMEN

Matière	:	Algorithmique et Programmation 2	Classes	:	TI1x
Enseignants	:	A. Ben Salem, J. Bokri, A. Elamraoui, O. Henia, Y. Landolsi,			
Date	:	17/06/2021	Documents	:	Non autorisé
Durée	:	01h30	Nbre pages	:	3

QCM (3pts)

Veuillez choisir **la** réponse exacte à chaque question **(ne pas rendre l'énoncé de l'examen).**

Question 1/
Pour lire au clavier 2 réels à placer dans les éléments t[1] et t[3] d'un tableau, on peut écrire:

A.
```
scanf("%f %f", *(t+1),
*(t+3));
```
B.
```
scanf("%f %f", (*t)+1,
*(t)+3);
```
C.
```
scanf("%f %f", t+1,
t+3);
```
D.
```
scanf("%f %f", t[1],
t[3]);
```

Question 2/
Pour afficher le premier et le deuxième caractère d'une chaine ch, on peut écrire:

A.
```
printf("%c %c", *ch,
*ch+1);
```
B.
```
printf("%c %c", *ch+1,
*ch+2);
```
C.
```
printf("%c %c", ch,
ch+1);
```
D.
```
printf("%c %c", *ch,
*(ch+1));
```

Question 3/
Soit la Structure suivante

```
struct Etudiant
{ float moyenne ;
int age ;
char nom[20] ;
} ;
struct Etudiant *E1 ;
```

Pour saisir le nom, on peut écrire

A. `scanf("s", &E1.nom);`
B. `scanf("s", E1->nom);`
C. `scanf("s", (*E1).nom);`

Question 4
Pour accéder au dernier élément d'un tableau Tab de 5 entiers, on peut écrire:

A. `printf("%d", *(Tab+5));`
B. `printf("%d", Tab[5]);`
C. `printf("%d",*(Tab+4));`
D. `printf("%d",*Tab+4);`

Question 5
Quel est l'affichage correspondant au code ci-dessous ?

```
int x=4;
int y=6;
int *p1=&x, *p2=&y;
*p2 = 10;
 *p1= 5;
 x+=2;
y-=3;
printf("%d %d %d %d ", x, y,
*p1, *p2);
```

A. 6 3 10 5
B. 7 7 10 5
C. 7 7 7 7
D. 4 6 10 5

Question 6

Soit l'entête de la fonction suivante:

```
void f1(int *x , int *y,float
*z,char *ch);
```

et les variables suivantes:

```
int *p1,*p2,a,b;
char nom[20];
float *p3,c;
```

Les appels possibles de f1 sont :

A. `f1(*p1 ,*p2,*p3,*nom);`
B. `f1(&p1 &p2,&p3,&nom);`
C. `f1(p1 p2,p3,nom);`
D. `f1(a,b,c,nom);`
E. `f1(&a,&b,&c,nom);`

Exercice 1 : (6 pts)

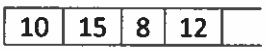
Une file est une structure de données de type **FIFO** (First In First Out) ce qui veut dire que les premiers éléments ajoutés à la file seront les premiers à en être retirés.
Une pile est une structure de données de type **LIFO** (Last In First Out) : le dernier entré est le premier sorti.

En utilisant, les opérations de base d'une pile (**creerPile, videPile, sommetPile, empiler** et **depiler**) et celles d'une file (**creerFile, videFile, sommetFile, emfiler** et **defiler**), *sans les implémenter*, écrire les fonctions suivantes :

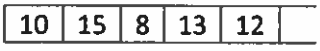
1. **Somme (n)** : cette fonction calcule la somme de 1 à n, avec n est un entier positif non nul, à l'aide d'une seule pile. **(2 pts)**
   Exemple :

Pile p Somme(5) retourne 15

2. **Inserer (F, x, pos)** : cette fonction permet d'insérer, dans la file **F**, un entier **x** à une position **pos** (sachant que **pos** est le nombre de pas à partir du sommet de la file), en utilisant seulement des piles intermédiaires. **(2 pts)**
   Exemple :



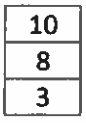
Inserer (F, 13, 4) donne la file suivante



3. **Supprimer (P, x)** : cette procédure permet de supprimer, de la pile **P**, un entier **x** en utilisant seulement une file et une pile intermédiaires. **(2 pts)**

Exemple :

Pile P Supprime (P,25) donne la pile :



Exercice 1 (11 pts)

1. Qu'est-ce qu'un arbre binaire de recherche? **(0.5pt)**
2. Ecrire les déclarations des structures nécessaires pour un arbre binaire de recherche où le champ information du nœud est de type entier. **(1pt)**
3. Dessiner l'arbre binaire de recherche **A** après avoir y inséré successivement les valeurs suivantes : **21-30-15-25-10-35-31-5-12-33-13-3 (1pt)**
4. Ecrire les éléments de cet arbre de 3 manières :
    a. Parcours préfixé **(0.5pt)**
    b. Parcours postfixé **(0.5pt)**
    c. Parcours infixé **(0.5pt)**
5. Ecrire la fonction **Parcours_infixe(...). (0.5 pt)**
6. Ecrire une fonction **itérative insererABR** qui permet d'insérer un entier x dans l'arbre de telle sorte que l'arbre reste ordonné.**(1.5pts)**
7. Ecrire une fonction récursive **rechercherABR** permettant de rechercher un entier **x** dans un arbre binaire de recherche, de retourner 1 s'il existe et 0 sinon. **(1.5pts)**
8. Ecrire une fonction **tailleArbre** permettant de déterminer la taille d'un arbre (le nombre de nœuds). **(1pt)**
9. Ecrire une fonction **CompetArbre** permettant de tester si un arbre est complet (un arbre est dit complet si chaque nœud possède un fils gauche et un fils droit) **(1pt)**
10. Ecrire la fonction principale main où vous créer un menu pour gérer un arbre binaire de recherche **(1.5pts)**

Ministère de l'Enseignement Supérieur,
de la Recherche Scientifique, et de la Technologie

Direction Générale des Etudes Technologiques
Institut Supérieur des Etudes Technologiques
de Bizerte

Département : Technologies de l'informatique		Année : 2020 – 2021 Semestre : 2
Module d'enseignement : Programmation Web 2	**Classe : TI1X**	Durée : 1h30
Equipe pédagogique : D. Amara, A. Gafsi, M. Hamza, I. Jemmali, H. Hedhly		☐ Devoir surveillé ☒ Examen
Nom : **Prénom :**		☐ Documents autorisés ☒ Documents Non autorisés

Énoncé

On se propose de développer une application pour gérer les tâches à faire « **TODO APP** ». Le projet contient un fichier HTML et 3 fichiers JavaScript.

Les codes des fichiers sont donnés entièrement ou partiellement dans l'annexe (voir figures 3, 4, 5 et 6). Ces codes seront utilisés pour formuler les réponses.

Figure 1 Arborescence des fichiers

L'application contient une seule page `index.html`. Cette page se divise en deux zones (voir figure 2) :

1. Zone 1 : formulaire (input + bouton) permettant d'ajouter de nouvelles tâches.
2. Zone 2 : une liste contenant les tâches qui sont déjà saisies.

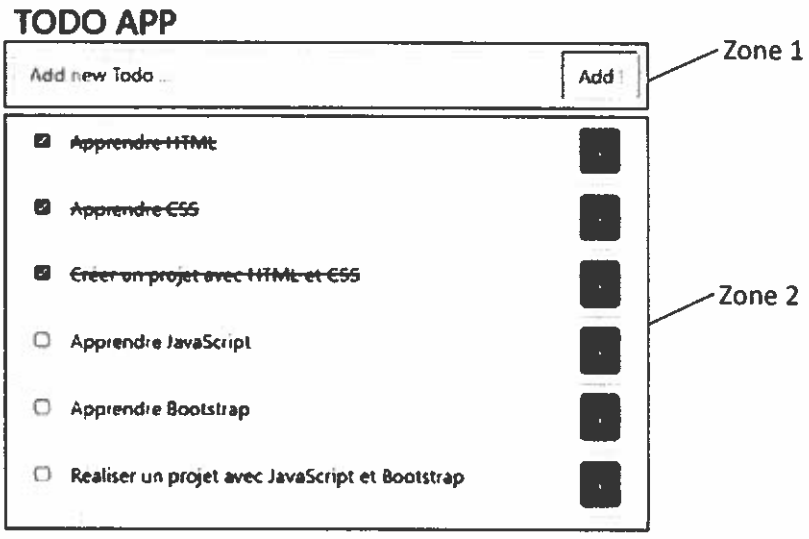
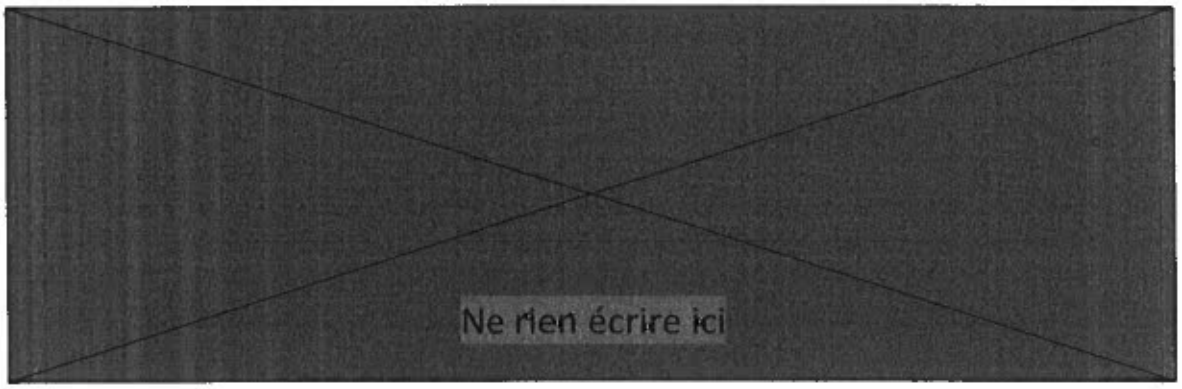


Figure 2 index.html

Important : Le code de la page `index.html`, donné dans figure 3 de l'annexe, **ne doit pas être changé** et **vous devez l'utiliser tel qu'il est dans vos réponse**.



Travail à faire

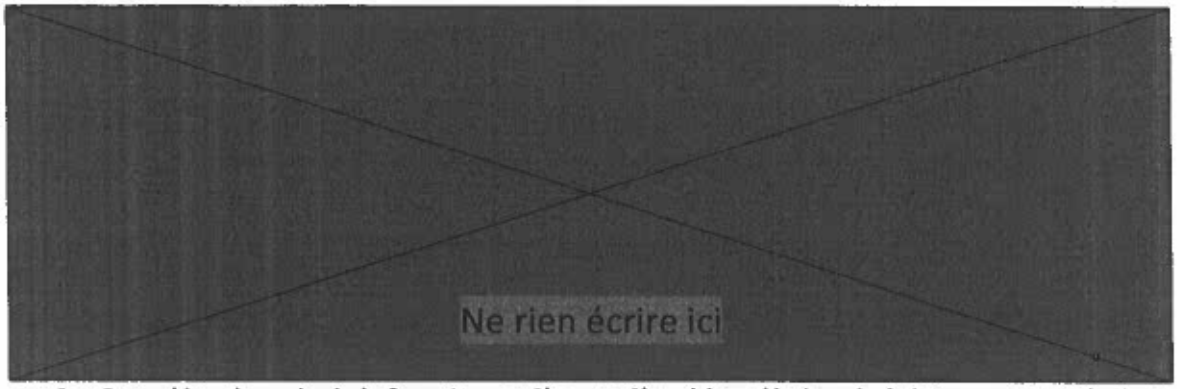
Important : Pour chacune des questions suivantes des commentaires sont présents dans le code afin de vous guider dans les réponses.

1. Compléter le code principal dans le fichier `main.js` (voir figure 6).

```
/************************** CODE PRINCIPAL ************************/
// Sélectionner la liste ul depuis le DOM
.....
// Sélectionner le bouton d'ajout depuis le DOM et y ajouter un écouteur de
clique
.....
.....
// Lire le contenu du LocalStorage. S'il c'est vide initier un tableau vide
toDos = loadDataFromLocalStorage('toDos') ?? [];
displayToDos();
```

2. Compléter le code de la fonction `displayToDos()` dans le fichier `main.js` (figure 6) permettant de lire le contenu depuis le `LocalStorage` et l'ajouter dans la liste `ul`.

```
function displayToDos() {
    ulList.innerHTML = ""; // Supprime le contenu de la liste avant l'affichage
 // tester si le tableau est vide alors afficher dans ul un seul li : No Todos
    if ( ..... ) {
    .....
    .....
    .....
    .....
  } else { // Sinon lire le contenu du tableau et l'injecter dans la ul
        toDos.forEach(.....
      .....
      .....
  }
  // ajouter des event listeners pour capter le changement des checkbox et le
  // clique sur les boutons de suppression
    installEventListener('input[type=checkbox]', 'change', onChangeCheckbox);
    installEventListener('button.delButtons', 'click', onClickDeleteButton);
}
```

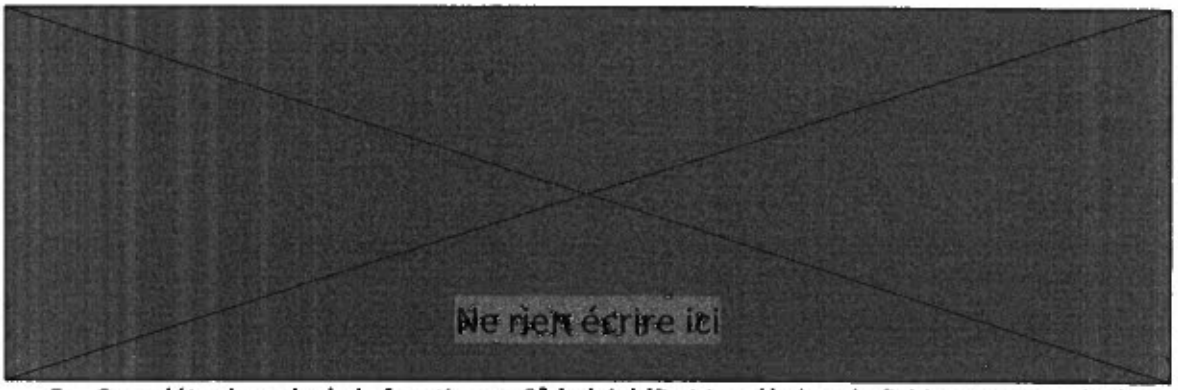


3. Compléter le code de la fonction onChangeCheckbox() dans le fichier events.js (voir figure 5) qui sera appelé si le checkbox d'une des tâches a été changé et donc modifier la valeur de la propriété completed de la tâche en question.

```
function onChangeCheckbox() {
  // parcourir la liste des tâches
    for (.....) {
    // vérifier dans le tableau l'élèment ayant l'id correspondant à celui de la
    // case coché
        if(.....) {
            // changer la valeur de completed à l'inverse
            .....
            // enregister la liste dans le LocalStorage
            .....
            // appeler la fonction displayToDos() pour actualiser l'affichage
            .....
        }
    }
}
```

4. Compléter le code de la fonction onClickDeleteButton() dans le fichier events.js (voir figure 5) appelée si on clique sur un des boutons de suppression.

```
function onClickDeleteButton() {
  if (confirm('Voulez-vous vraiment supprimer cette tache ?')) {
    for (.....) { // parcourir les tâches
      if(toDos[i].id == this.title) {
        // supprimer la case de l'élèment en question
        .....
        // enregistrer la liste dans le LocalStorage
        .....
        // appeler la fonction displayToDos() pour actualiser l'affichage
        .....
        alert('Tache supprimée avec succès!');
      }
    }
  }
```



5. Compléter le code de la fonction onClickAddButton() dans le fichier events.js (voir figure 5) appelée si on clique sur le bouton d'ajout.

```
function onClickAddButton() {
  // récupérer ce que l'utilisateur a saisie dans l'input
  .....
  if (.....) { // tester sur la longueur du saisie dans l'input
    const newTodo = ..... ; // créer un objet vide
    newTodo.id = toDos.length === 0 ? 1 : toDos[toDos.length-1].id + 1;
    // affecter dans la propriété title la valeur saisie dans l'input
    .....
    newTodo.completed = false; // nouvelle tâche est par défaut not completed
    // insérer la nouvelle tâche dans le tableau des tâches
    .....
    // enregistrer la liste dans le LocalStorage
    .....
    displayToDos();
    // vider l'input de la valeur saisie
    .....
  } else {
    // afficher une alerte qui demande de bien saisir le titre
    .....
  }
}
```

Fonctionnement

Toutes les tâches sont stockées dans un `LocalStorage` nommé **`toDos`** (voir figure 7). Lors de l'ouverture de la page, nous récupérons ces tâches avec JavaScript pour les injecter dans la liste `ul`. Si la liste dans le `LocalStorage` est vide on affiche « No Todos ! ».

Chaque tâche a trois propriétés :

- `id` : pour distinguer les différentes tâches
- `title` : Le titre de la tâche qui sera affiché
- `completed` : pour vérifier si la tâche est déjà terminée ou non.

Si la tâche est déjà terminée (`completed : true`), on affiche le `checkbox` qui lui correspond `checked` et son titre barré. Si on coche ou on décoche le `checkbox`, on change, dans le `LocalStorage`, la valeur de `completed` de `true` vers `false` et inversement.

Pour chaque tâche on affiche un bouton permettant de supprimer la tâche depuis le `LocalStorage`. Avant la suppression, on vérifie avec l'utilisateur s'il souhaite bien supprimer la tâche, s'il confirme, on affiche une alerte pour l'informer que la suppression a été faite.

Quand l'utilisateur saisie le titre de la tâche dans l'input de la zone 1 et clique sur le bouton d'ajout, on vérifie que la longueur du titre saisie est supérieur à 2 caractères. Si c'est le cas on enregistre la tâche dans le `LocalStorage`, on réaffiche la nouvelle liste et on vide le contenu de l'input. Si non on lui affiche une alerte qu'il doit saisir un titre.

Annexes

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link href="bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="js/utilities.js"></script>
    <script src="js/events.js"></script>
    <script src="js/main.js" defer></script>
    <title>Todo-List</title>
</head>
<body>
  <div class="container py-3">
    <header>
      <h2>TODO APP</h2>
    </header>
    <main>
      <div class="input-group mb-3">
        <input type="text" class="form-control" placeholder="Add new Todo ...">
        <button class="btn btn-outline-secondary" type="button" id="addButton">
              Add !
        </button>
      </div>
      <ul class="list-group"></ul>
```

```
    </main>
  </div>
</body>
</html>
```

Figure 3 Code HTML de la page index.html

```
/**
 * Read data from LocalStorage and convert it
 * @param {string} storageName: Key of LocalStorage property
 * @returns data from LocalStorage
 */
function loadDataFromLocalStorage(storageName) {
    const data = JSON.parse(localStorage.getItem(storageName));
    return data;
}
/**
 * Convert and save data to LocalStorage
 * @param {string} storageName: Key of LocalStorage property
 * @param {any} data: Data to save into LocalStorage
 */
function saveDataToLocalStorage(storageName, data) {
    const jsonData = JSON.stringify(data);
    localStorage.setItem(storageName, jsonData);
}
/**
 * Install an event listener on an element with a given selector
 *
 * @param {string} selector: the selector to used to get the element from DOM
 * @param {string} ev: name of the event to install
 * @param {function} eventHandler: name of the event handler function
 */
function installEventListener(selector, ev, eventHandler) {
    const elements = document.querySelectorAll(selector);
    elements.forEach(elem => elem.addEventListener(ev, eventHandler));
}
```

Figure 4 Code du fichier utilities.js

```
function onChangeCheckbox() {
    . . .
}
function onClickDeleteButton() {
    . . .
}
function onClickAddButton() {
    . . .
}
```

Figure 5 Code du fichier events.js

```
/*********************************************************************
 ************************ DONNEES GLOBALES ************************
 *********************************************************************/
let toDos;
let ulList;
let addButton;
/*********************************************************************
 ************************** FONCTIONS *****************************
 *********************************************************************/
 /**
  * Permet de créer un input de type checkbox
  * @param {bool} checked
  * @param {integer} id
  * @returns checkbox
  */
 function createCheckBox(checked, id) {
    const checkBox = document.createElement('input');
    checkBox.setAttribute('type', 'checkbox');
    checkBox.setAttribute('title', id);
    checked ? checkBox.setAttribute('checked', 'checked') : '';
    checkBox.classList.add('me-3');
    return checkBox;
 }
 /**
  * Permet  de créer un nouveau bouton de suppression
  * @param {integer} id : de l'élèment à supprimer
  * @returns buttonElement
  */
 function createDeleteButton(id) {
     const delBtn = document.createElement('button');
     delBtn.innerHTML = "x";
     delBtn.setAttribute('title', id);
     delBtn.className = "btn btn-danger float-end delButtons";
     return delBtn;
 }
 /**
  * Permet de créer un li avec la classe list-group-item
  * @returns li
  */
 function createListGroupItem() {
     // créer un nouvel élément li
    const li = document.createElement('li');
    // ajouter la classe list-group-item à l'élément li
    li.className = "list-group-item";
    return li;
 }
 /**
  * Insère un todo dans la liste des toDos
  * @param {object} todo
```

```
 */
function injectToDoInDom(todo)  {
    // créer un nouvel élément li
    const li = createListGroupItem();
    // ajouter un checkbox comme premier fils à l'élément li
    li.appendChild(createCheckBox(todo.completed, todo.id));
    // ajouter le titre du todo comme deuxième  fils à l'élément li
    const title = document.createTextNode(todo.title);
    // vérifier si le Todo est completed, le titre sera alors barré
    if (todo.completed) {
        const stroke =  document.createElement('del');
        stroke.appendChild(title);
        li.appendChild(stroke);
    } else {
        li.appendChild(title);
    }
    // ajouter le boutton de suppression comme deuxième  fils à l'élément li
    li.appendChild(createDeleteButton(todo.id));
    // ajouter l'élément li comme premier fils à la ulList
    ulList.appendChild(li);
}
/**
 *  Affiche le contenu de la liste des Todo depuis le localstorage
 */
function displayToDos() {
   . . . // code à compléter
}
/*********************************************************************
 ************************ CODE PRINCIPAL ************************
 *********************************************************************/
. . . // code à compléter
toDos = loadDataFromLocalStorage('toDos') ?? [];
displayToDos();
```

Figure 6 Code du fichier main.js

Key	Value
toDos	[{"id":1,"title":"Apprendre HTML","completed":true},{"id":2,"title":"App

```
▼[{id: 1, title: "Apprendre HTML", completed: true}, {id: 2, title: "Apprendre CSS", compl
  ▶0: {id: 1, title: "Apprendre HTML", completed: true}
  ▶1: {id: 2, title: "Apprendre CSS", completed: true}
  ▶2: {id: 3, title: "Créer un projet avec HTML et CSS", completed: true}
  ▶3: {id: 4, title: "Apprendre JavaScript", completed: false}
  ▶4: {id: 5, title: "Apprendre Bootstrap", completed: false}
  ▶5: {id: 6, title: "Réaliser un projet avec JavaScript et Bootstrap", completed: false}
```

Figure 7 Contenu du LocalStorage

ISET Bizerte		Année Universitaire : 2020/202021 Session : Juin 2021	
Département Technologie de l'information	***EXAMEN***	Statistiques et probabilités	
Enseignant : *A. BESSAOUDIA*		Groupes : TI11, 12, 13,14,15	
		Durée : 1H30	Nbre de pages : 1

Exercice 1 (12 points)

Le tableau suivant donne la distance de freinage d'un véhicule automobile roulant sur une route sèche, en fonction de sa vitesse :

Vitesse (km/h) Xi	40	50	60	70	80	90	100	110	120
Distance de freinage (m) Yi	8	12	18	24	32	40	48	58	72

1- Représenter le nuage de points associé à cette série statistique (X,Y)
2- Calculer la covariance de X et Y, interpréter.
3- Calculer le coefficient de corrélation entre X et Y, interpréter.
4- Déterminer la droite d'ajustement linéaire par la méthode des moindres carrés ordinaires
5- Estimer la distance de freinage d'un véhicule roulant à 130 km/h.
6- Estimer la vitesse d'un véhicule qui freine sur 100 mètres.

Exercice 2 (8 points)

Le tableau suivant donne le chiffre d'affaires Y (en milles dinars) réalisé au cours des 6 derniers mois par un site de vente en ligne en fonction du nombre de commandes reçues X.

Nombre de commandes Xi	6400	8300	9125	9600	10050	12000
Chiffre d'affaires Yi	250	320	335	350	370	400

1- Représenter le nuage de points associé à cette série statistique (X,Y)
2- Déterminer la droite d'ajustement linéaire par la méthode de Mayer
3- Déterminer le chiffre d'affaires pour 15000 commandes reçues.

Direction Générale des Etudes Technologiques Institut Supérieur des Etudes Technologiques de Bizerte Département TI		
Matière : Recherche opérationnelle **Enseignants :** Sabri Hosni, M.Hnid	**Devoir de synthèse**	**Durée :** une heure et demie
Classe : TI1		**AU :** 20/21

Exercice1 :

Soit le **pl** suivant relatif à un problème de production qui comporte deux produits et qui consiste à maximiser le bénéfice global :

MAX=200X1+400X2

S/C : 2 X1+6X2 ≤ 900

4X1+2X2 ≤ 700

X1+X2 ≤ 200

X1 ≥ 0 ; X2 ≥ 0

1. Donnez une interprétation économique de ce modèle de programme linéaire.
2. Trouvez la solution optimale par la méthode graphique.
3. Trouvez la solution optimale par la méthode de simplexe.
4. Que devient la solution optimale trouvée si le coefficient de X2 change de 400 à 600

Exercice2 :

Soit le **PL** suivant :

MAX=2X1+6X2

S/C : 4X1+12X2 ≤ 60

X1 ≤ 10

X2 ≤ 4

X1≥0 ; X2≥0

1. Résolvez graphiquement ce pl.
2. Trouvez la solution par la méthode de simplexe
3. Quelle est la nature de la solution trouvée.

Ministère de l'enseignement supérieur et de la recherche scientifique
Direction Générale des Etudes Technologiques
Institut Supérieur des Etudes technologiques de Bizerte
Département Technologies de l'Informatique

Matière	: Systèmes d'exploitation	**Classes**	: TI1x
Enseignants	: A. Aouini, A. Gharbi	**Durée**	: 01h30
Date	: Juin 2021	**Documents**	: Non autorisés
		Nbr pages	: 3 pages

Questions de cours (4 points)

1. Expliquer le principe de l'algorithme d'ordonnancement à priorités.

2. Expliquer, brièvement, le principe de swapping.

3. Citer les deux types de fragmentation. Expliquer comment elle engendre un problème et par quel moyen est résolu.

4. Citer les différences entre une mémoire paginée et une mémoire segmentée.

Exercice1 : Algorithmes d'ordonnancement (4.5 points)

On présente la table d'exécution des processus.

Processus	*Date d'arrivée*	*Temps de traitement*	*Priorité*
P_1	0	5	4
P_2	3	4	2
P_3	3	2	6
P_4	5	4	3

1. On suppose utiliser l'algorithme d'ordonnancement SJF pour présenter le diagramme d'exécution des processus

2. Utilisez l'ordonnancement à priorités pour présenter le diagramme d'exécution des processus .On suppose qu'un nombre de priorité élevé correspond à une priorité plus importante.

a) Priorité sans préemption

b) Priorité avec préemption

3. Calculez pour les algorithmes d'ordonnancement étudiés :

- Le temps de Rotation de chaque processus et le Temps de Rotation Moyen
- Le temps d'Attente de chaque processus et le Temps d'Attente Moyen

a) SJF

b) Priorité sans préemption
c) Priorité avec préemption

Exercice 2 : Pagination (4 points)

Dans un système paginé, les pages font 256 octets et on autorise chaque processus à utiliser au plus 3 cadres de la mémoire centrale. On considère la table des pages suivante du processus P1 :

Page	0	1	2	3	4	5	6	7
Cadre	6	2	1	3	5	0	4	7
Présence	Oui	Non	Oui	Non	Non	Non	Oui	Non

1. Quelle est la taille de l'espace d'adressage du processus P1 ?

2. De combien de mémoire vive dispose le processus ?

3. Calculez les adresses réelles (c'est-à-dire physique) correspondant aux adresses virtuelles suivantes (signalez, éventuellement, les erreurs d'adressage) : 250, 556, 1588, 2062.

Exercice 3 : Algorithmes de Remplacement de Pages (4 points)

Un programme P possède un espace virtuel de 600 octets. On considère la suite des adresses

virtuelles qui suit :

44, 223, 345, 520, 356, 447, 310, 11, 21, 6, 284, 236, 398, 599, 225, 174, 52, 388, 156, 266.

1. Donner la suite des numéros de pages référencées, sachant qu'elles comportent 100 Octets et le numéro des pages commence par 0.

Le programme dispose de 300 octets en mémoire centrale.

2. Calculer le nombre de défauts de page (en supposant que la mémoire est initialement vide) pour les algorithmes :

a) FIFO (First In First Out)

b) LRU (Least Recently Used)

c) Optimal

Exercice 4 au choix (<u>choisir un seul exercice</u>) (3.5 points)

<u>Choix 1</u>: Allocation dynamique de la mémoire centrale

Soit un système qui utilise l'allocation par partitions variables.

La figure illustre l'état de la mémoire centrale à l'instant t (les zones libres et occupées)

En fonction des évènements suivants :

- Arrivée du processus P6 (10ko)
- Départ du processus P4
- Arrivée du processus P7 (15ko)
- Départ du processus P5
- Arrivée du processus P8 (30ko)

1. Représenter l'allocation de la mémoire centrale avec l'algorithme First Fit.

2. Représenter l'allocation de la mémoire centrale avec l'algorithme Best Fit.

Choix 2 : I-node

On considère un disque dur utilisant un système de fichiers où l'information concernant les blocs de données est accessible à partir de l'i-node du fichier concerné.

On suppose que :

- L'i-node contient 12 pointeurs directs sur les blocs et un pointeur indirect.
- Un pointeur occupe 4 octets
- Les blocs de données sont de taille fixe 1Ko

1. Déterminer la taille maximale (en octet) d'un fichier pour ce système.

2. Représentez sur un schéma l'allocation des blocs de données d'un fichier de taille
   a)11 KO
   b)100 KO

ISET BIZERTE

Durée : 01h30

Niveau : 1ére année
Section : TI

EXAMEN

Fondement des Réseaux

Enseignantes : I.ABBES, Z.LANGAR, I.TURKI, J.BOKRI, A.GHARBI

Année Universitaire: 2020/2021

Nombre de pages : 7 pages

Nom :..... PRENOM :..... CIN :.....
CLASSE :.....

Exercice 1 (4 pts):

1. Donner les fonctionnalités de la couche transport

.....
.....
.....
.....

2. Quel protocole de la couche 4 est utilisé pour les applications vidéo ?

.....
.....
.....
.....

3. Citer trois améliorations apportées par l'adressage IPV6

.....
.....
.....
.....
.....
.....

4. Compléter le tableau suivant

Adresse IPV6	Format compressé
FE80:0000:0000:0000:0000:4CFF:FE4F:4F50	
2001:0688:1F80:0000:0203:FFFF:4C18:00E0	
3CD0:0000:0000:0000:0040:0000:0000:0CF0	

NE RIEN ECRIRE ICI

1. Déterminer le masque de sous-réseau permettant de gérer le nombre d'adresses IP nécessaires au LAN 1 du switch Sw-1?

.....

.....

.....

.....

2. Déterminer le masque de sous-réseau permettant de gérer le nombre d'adresses IP nécessaires au LAN 2 du switch Sw-2?

.....

.....

.....

.....

.....

3. Déterminer le masque de sous-réseau permettant de gérer le nombre d'adresses IP nécessaires au LAN 3 du switch Sw-3?

.....

.....

.....

.....

4. Déterminer le masque de sous-réseau permettant de gérer le nombre d'adresses IP nécessaires au LAN 4 du switch Sw-4?

.....

.....

.....

.....

.....

.....

5. Déterminer le masque de sous-réseau permettant de gérer le nombre d'adresses IP nécessaires à la connexion entre R1 et R2 ?

.....

.....

.....

6. Déterminer le masque de sous-réseau permettant de gérer le nombre d'adresses IP nécessaires à la connexion entre R2 et R3 ?

.....

.....

.....

7. Documenter la table de sous-réseau en divisant le réseau **172.68.0.0/16** en fonction du nombre d'hôtes par sous-réseau, et ce comme suit :
   a. Utilisez le premier sous-réseau pour accueillir le plus grand LAN.
   b. Utilisez le deuxième sous-réseau pour accueillir le deuxième LAN le plus grand.
   c. Utilisez le troisième sous-réseau pour accueillir le troisième LAN le plus grand.
   d. Utilisez le quatrième sous-réseau pour accueillir le quatrième LAN le plus grand.
   e. Utilisez le cinquième sous-réseau pour gérer la connexion entre R1 et R2.
   f. Utilisez le sixième sous-réseau pour gérer la connexion entre R2 et R3.

NE RIEN ECRIRE ICI

Table des sous-réseaux

Description du sous-réseau	Adresse réseau/CIDR	Première adresse d'hôte utilisable	Adresse de diffusion
LAN.... attribué au SW-.....			
LAN.... attribué au SW-.....			
LAN.... attribué au SW-.....			
LAN.... attribué au SW-.....			
Liaison entre R1 et R2			
Liaison entre R2 et R3			

8. Documenter la table d'adressage ci-dessous, en respectant les règles suivantes :

a. Attribuez les premières adresses IP utilisables à R1 pour la liaison LAN et la liaison WAN.

b. Attribuez la première adresse IP utilisable à R2 pour la liaison LAN, la première adresse IP utilisable à la liaison WAN avec R3 et la dernière adresse IP utilisable à la liaison WAN avec R1.

c. Attribuez les premières adresses IP utilisables à R3 pour les liaisons LAN et la dernière adresse IP utilisable à la liaison WAN avec R2.

d. Attribuez les dernières adresses IP utilisables aux hôtes.

NE RIEN ECRIRE ICI

Table d'adressage

Appareil	Interface	Adresse IP	Masque de sous-réseau	Passerelle par défaut
R1	G0/0			N/A
	S0/0/0			N/A
R2	G0/0			N/A
	S0/0/0			N/A
	S0/0/1			N/A
R3	G0/0			N/A
	G0/1			N/A
	S0/0/0			N/A
Host-A	Carte réseau			
Host-B	Carte réseau			
Host-C	Carte réseau			
Host-D	Carte réseau			