

EXAMEN TP

Semestre : 1 ☐ 2 ☒

Session : Principale ☒ Rattrapage ☐

Module : Programmation Procédurale II

Enseignant(s) : Equipe algorithmique

Classe(s) : 1A

Documents autorisés : OUI ☒ NON

Nombre de pages : 3

Date : 01/03/2016 Heure : 9H00

Durée : 1H30

Remarques :

- Les cours, TD et TP de C sont autorisés.
- IL est impératif d'utiliser **la programmation modulaire**.

Enoncé :

Dans un atelier de fabrication de pièces en plastique les travaux sont représentés par une liste chaînée. Un travail étant caractérisé par un numéro (entier), un code (chaîne de caractère) et un temps opératoire (entier) .

Le chef de l'atelier doit chaque matin préparer la liste des travaux de la journée. Ces travaux seront représentés par une liste simplement chaînée.

A la fin de la journée, le chef de l'atelier voudra avoir une pile contenant les travaux classés selon le temps opératoire de chaque travail.

Pour ce faire :

- Déclarer la structure de données travail
- Déclarer la structure de données cellule
- Déclarer le type liste comme pointeur sur le type cellule
- Déclarer le type pile

Développer :

1. Les fonctions qui permettent de saisir et d'ajouter un travail à la fin de la liste des travaux.

travail saisir()

liste ajouter_travail(liste L, travail T)

2. La fonction qui permet d'afficher la liste des travaux et de retourner et afficher le nombre des travaux de la liste.

int afficher_Liste(liste L)

3. Si, au milieu de la journée, le chef désire planifier un nouveau travail, il doit l'insérer avant le premier travail rencontré ayant le même code opératoire que celui à insérer.

Si le code opératoire du travail à ajouter n'existe pas déjà dans la liste des travaux la fonction devra afficher un message.

liste inserer_Travail(liste L, travail T)

4. La fonction qui permet de supprimer le premier travail rencontré dont le numéro est passé en paramètres de la fonction.

liste supprimer_Travail(liste L, intnumero)

5. La fonction qui permet de retourner le travail qui a le plus long temps opératoire

travail max_temps_op(liste L)

6. La fonction qui permet d'empiler tous les travaux de la liste en commençant par le travail qui a le plus long temps opératoire. La fonction permettra aussi de supprimer de la liste chaque élément empilé.

*pile empiler_Travaux(pile P, liste *L)*

NB : La fonction doit faire appel aux deux fonctions précédentes.

7. La fonction qui permet de consulter le sommet de la pile

void consulter_pile(pile P)

8. Implémenter la fonction **main** qui permet, grâce à un menu de :

- Ajouter un travail
- Afficher la liste des travaux
- Insérer un nouveau travail
- Supprimer un travail
- Empiler les travaux par temps opératoire
- Consulter la pile

Grille d'évaluation

Nom :Classe :

Travail demandé		Exécution (**)		Algorithmique(*)		Total	
.h .c							0.5
Définition de la structure travail							0.5
Définition de la structure cellule							0.5
Définition de type liste							0.25
Définition de type pile							0.25
Ajouter_travail			0.5		1		1.5
Afficher_Liste	Avec calcul nbre		0.5		1		1.5
	Sans calcul (-0.5)						
Insérer_travail (insertion simple -3)			1.5		2.5		4
Supprimer_travail			1		1,5		2,5
Max_temps_op			1		1		2
empiler			1		2.5		3.5
consulter			0.5		0.5		1
Main+menu			2		-		02
Total :							20

(*) Explication

(**) LA NOTATION DE LA PARTIE EXECUTION EST BINAIRE.