

I. Liste chaînées

Enregistrement *élément* :

 | valeur : entier,
 | suivant : lien sur élément

Fin enregistrement

Procédure affiche(*élém*)

 | **Spécification** :

 | *Entrée* : un lien sur un élément
 | *Sortie* : rien

 | **Si** *élém* $\neq \emptyset$ **alors**

 | **Afficher** *élém* \rightarrow valeur
 | affiche(*élém* \rightarrow suivant)

 | **Fin si**

Procédure append(*liste*, *val*)

 | **Spécification** :

 | *Entrée* : un lien sur un élément, un entier
 | *Sortie* : un lien sur un élément

 | **Variables** :

 | *élém* : lien sur élément

 | **Si** *liste* = \emptyset **alors**

 | *élém* \leftarrow nouveau()
 | *élém* \rightarrow valeur \leftarrow *val*
 | *élém* \rightarrow suivant $\leftarrow \emptyset$

 | **Retourner** *élém*

 | **Sinon**

 | *liste* \rightarrow suivant \leftarrow append(*liste* \rightarrow suivant, *val*)

 | **Retourner** *liste*

 | **Fin si**

Procédure count(liste, val)

Spécification :

Entrée : un lien sur un élément, un entier

Sortie : un entier

Si liste = \emptyset **alors**

Renvoyer 0

Sinon si liste \rightarrow valeur = val **alors**

Renvoyer 1 + count(liste \rightarrow suivant, val)

Sinon

Renvoyer count(liste \rightarrow suivant, val)

Fin si

On suppose la liste triée par ordre décroissant.

Procédure place(liste, val)

Spécification :

Entrée : un lien sur un élément, un entier

Sortie : un lien sur un élément

Variables :

 élément : lien sur élément

Si liste = \emptyset **alors**

 élément \leftarrow nouveau()

 élément \rightarrow valeur \leftarrow val

 élément \rightarrow suivant \leftarrow \emptyset

Retourner élément

Sinon si liste \rightarrow valeur \geq val **alors**

 liste \rightarrow suivant \leftarrow place(liste \rightarrow suivant, val)

Retourner liste

Sinon

 élément \leftarrow nouveau()

 élément \rightarrow valeur \leftarrow val

 élément \rightarrow suivant \leftarrow liste

Retourner élément

Fin si

On suppose la liste triée par ordre croissant.

Procédure place(liste, val)

Spécification :

Entrée : un lien sur un élément, un entier

Sortie : un lien sur un élément

Variables :

 élé_m : lien sur élément

Si liste = \emptyset **alors**

 élé_m \leftarrow nouveau()

 élé_m \rightarrow valeur \leftarrow val

 élé_m \rightarrow suivant \leftarrow \emptyset

Retourner élé_m

Sinon si liste \rightarrow valeur \geq val **alors**

 élé_m \leftarrow nouveau()

 élé_m \rightarrow valeur \leftarrow val

 élé_m \rightarrow suivant \leftarrow liste

Retourner élé_m

Sinon

 liste \rightarrow suivant \leftarrow place(liste \rightarrow suivant, val)

Retourner liste

Fin si

Procédure remove(liste, val)

Spécification :

Entrée : un lien sur un élément, un entier

Sortie : un lien sur un élément

Variables :

 suivant : lien sur élément

Si liste = \emptyset **alors**

Renvoyer liste

Sinon si liste \rightarrow valeur = val **alors**

 suivant \leftarrow liste \rightarrow suivant

 libère(liste)

Renvoyer suivant

Sinon

 liste \rightarrow suivant \leftarrow remove(liste \rightarrow suivant, val)

Renvoyer liste

Fin si

II. Traitement d'ordres d'usinage

Enregistrement pièce :

 type : chaîne,

 durée : entier,

 date_fin : entier,

 suivant : lien sur pièce

Fin enregistrement

II.1. Traitement dans l'ordre du fichier

Procédure lecture(fichier, liste)

Spécification :

Entrée : une chaîne, un lien sur une pièce

Sortie : un lien sur une pièce

Variables :

 élé_m : lien sur pièce

Si <fin de la lecture du fichier fichier> **alors**

Renvoyer liste

Si non

 élé_m ← nouveau()

 élé_m → type ← lire(fichier)

 élé_m → durée ← lire(fichier)

 élé_m → date_fin ← lire(fichier)

Renvoyer élé_m

Fin si

Procédure traite(liste, temps_passé)

Spécification :

| *Entrée* : un lien sur une pièce, et un entier

| *Sortie* : rien

Variables :

| retard : entier

Si liste $\neq \emptyset$ **alors**

| retard \leftarrow liste \rightarrow date_fin - temps_passé

| **Afficher** « Traitement de la tâche [liste \rightarrow type] »

| On réalise la tâche liste \rightarrow type.

| **Si** retard > 0 **alors**

| | **Afficher** « Retard : [retard] »

| **Sinon**

| | retard $\leftarrow 0$

| **Fin si**

| **Renvoyer** traite(
| liste \rightarrow suivant,
| liste \rightarrow durée + temps_passé,
|) + retard

Fin si

Renvoyer 0

Algorithme principal

Variables :

| liste : lien sur une pièce

| retard_total : entier

| liste \leftarrow lecture("tâches.txt")

| retard_total \leftarrow traite(liste, 0)

II.2. Traitement dans l'ordre de date de fin

Procédure tri(liste)

|