

Projet IAP 2020/2021

Liste des instructions

Lexèmes

Définition. *< entier > : Un nombre positif, composé d'une suite de chiffres parmi 0,1...9, sans espace, ni virgule, ni aucun autre symbole. Sa valeur maximale est bornée à 10000 (inclus).*

Définition. *< mot > : Un mot, composé d'une suite de lettres parmi A,B..Z,a,b..z sans espace, ni virgule, ni aucun autre symbole (donc pas d'accentuation non plus). Sa longueur maximale est bornée à 35 caractères (inclus).*

Définition. *mot clé : Un mot réservé pour une instruction. Il ne peut être utilisé nulle part ailleurs. Noté en gras dans la suite.*

Spécialités

Instruction. ***developpe** < mot > < entier >*

Déclare une nouvelle spécialité de tâches qui sera proposée par l'entreprise. Le mot correspond au nom de la spécialité ; l'entier indique le prix de facturation d'une heure de travail pour cette spécialité.

Le système traite au maximum 10 spécialités.

Travailleurs

Instruction. ***embauche** < mot > < mot >*

Déclare un nouveau travailleur. Le premier mot correspond à son nom ; le second à sa spécialité. La spécialité doit avoir été précédemment déclarée. Chaque travailleur doit avoir un nom distinct.

Peut aussi être utilisée pour un travailleur déjà déclaré. Dans ce cas, une nouvelle spécialité est ajoutée pour ce travailleur. La spécialité doit avoir été précédemment déclarée.

Le système traite au maximum 50 travailleurs.

Clients

Instruction. ***demarche** < mot >*

Déclare un client. Chaque client doit avoir un nom distinct.

Le système traite au maximum 100 clients.

Commandes

Instruction. ***commande** < mot > < mot >*

Déclare une nouvelle commande. Le premier mot définit la commande. Chaque commande doit avoir un nom distinct. Le second mot correspond au nom du client effectuant la commande.

Le système traite au maximum 500 commandes.

Tâches

Instruction. ***tache** < mot > < mot > < entier >*

Déclare une tâche nécessaire à la réalisation d'une commande. Le premier mot désigne la commande ; le second, la spécialité concernée. L'entier indique le nombre d'heures nécessaires pour réaliser la tâche. Chaque commande peut avoir au maximum une tâche par spécialité.

Remarque : les tâches ne sont pas interdépendantes. Elles peuvent être réalisées en parallèle et avancer chacune à son rythme.

Affectation

L'affectation d'une tâche à un travailleur est faite automatiquement par le système. La règle dépend du sprint : soit le premier travailleur compétent connu, soit celui ayant reçu le moins de tâches, soit celui ayant la plus petite charge de travail restante.

Execution

Instruction. ***progression** < mot > < mot > < entier >*

Déclare l'avancement d'une tâche. Le premier mot désigne la commande ; le second la spécialité ; l'entier indique le nombre d'heures à ajouter à son avancement.

Instruction.

progression** < mot > < mot > < entier > **passe

*Lorsque le mot clé **passe** est ajouté, il indique qu'il faut, en plus du comportement ci-dessus, réaffecter la tâche.*

Remarque : il se peut que le même travailleur soit à nouveau sélectionné, s'il correspond encore au critère de choix lors de l'exécution de cette instruction.

Etat

Instruction. ***specialites***

Affiche la liste des spécialités, dans l'ordre de leur déclaration, ainsi que le coût associé.

Instruction. ***travailleurs** < mot >*

Affiche la liste des travailleurs compétents pour la spécialité choisie, par ordre de déclaration des travailleurs.

Instruction. ***travailleurs tous***

Affiche la liste des travailleurs pour toutes les spécialités. Les spécialités sont affichées dans l'ordre de leur déclaration ; pour une même spécialité, les travailleurs sont affichés par ordre de déclaration.

Instruction. ***client** < mot >*

Affiche la liste des commandes effectuées par le client choisi, selon leur ordre de déclaration.

Instruction. ***client tous***

Affiche la liste de toutes les commandes, par ordre de déclaration de client, et de commande.

Instruction. ***supervision***

Affiche la description de l'état d'avancement des commandes et des tâches associées, par ordre de commandes, et de spécialité.

Instruction. ***charge** < mot >*

Pour le travailleur choisi, affiche le nombre d'heures restantes pour l'accomplissement de chacune des tâches qui lui sont affectées. Les tâches sont affichées par ordre de commande, et de spécialité.

Facturation

Lorsqu'une commande est terminée, le système affiche automatiquement une facturation par spécialité effectivement utilisée.

Lorsque toutes les commandes sont terminées, le système affiche automatiquement une facturation globale.

Fin

Instruction. ***interruption***

Quitte brutalement le programme... L'usine est délocalisée.

Liste des sprints

Sprint 1 : La boucle des instructions

Chaque instruction provoque un affichage selon les formats ci-dessous. Chaque affichage est fait sur une ligne et se termine par un passage à la ligne, sauf pour le mot clé **passé** qui est fait sur deux lignes distinctes. L'instruction **interruption** provoque ensuite l'arrêt du programme ; les autres instructions n'ont pas d'effet.

developpe <i>mot entier</i>	<i>## nouvelle specialite "mot" ; cout horaire "entier"</i>
embauche <i>mot1 mot2</i>	<i>## nouveau travailleur "mot1" competent pour la specialite "mot2"</i>
demarche <i>mot</i>	<i>## nouveau client "mot"</i>
commande <i>mot1 mot2</i>	<i>## nouvelle commande "mot1", par client "mot2"</i>
tache <i>mot1 mot2 entier</i>	<i>## la commande "mot1" requiere la specialite "mot2" (nombre d'heures "entier")</i>
progression <i>mot1 mot2 entier</i>	<i>## pour la commande "mot1", pour la specialite "mot2" : "entier" heures de plus ont ete realisees</i>
progression <i>mot1 mot1 entier passé</i>	<i>## pour la commande "mot1", pour la specialite "mot2" : "entier" heures de plus ont ete realisees</i>
	<i>## une reallocation est requise</i>
specialites	<i>## consultation des specialites</i>
travailleurs <i>mot</i>	<i>## consultation des travailleurs competents pour la specialite "mot"</i>
travailleurs tous	<i>## consultation des travailleurs competents pour chaque specialite</i>
client <i>mot</i>	<i>## consultation des commandes effectuees par "mot"</i>
client tous	<i>## consultation des commandes effectuees par chaque client</i>
supervision	<i>## consultation de l'avancement des commandes</i>
charge <i>mot</i>	<i>## consultation de la charge de travail de "mot"</i>
interruption	<i>## fin de programme</i>

Versions : **alpha** les mots clé **passé** et **tous** ne sont pas traités (i.e. absents du jeu de test). **beta** Le mot clé **passé** n'est pas traité. **release** Tous les mots clés sont traités.

Guide technique

```
typedef enum {FAUX=0,VRAI=1} Boolean;
Boolean EchoActif = FAUX;
// Messages emis par les instructions -----
#define MSG_DEVELOPPE      "## nouvelle specialite \"%s\" ; cout horaire \"%d\"\\n"
#define MSG_INTERRUPTION  "## fin de programme\\n"
// Lexemes -----
#define LGMOT              35
#define NBCHIFFREMAX       5
typedef char Mot [LGMOT+1];
void get_id(Mot id){
    scanf("%s", id);
    if (EchoActif) printf(">>echo %s\\n", id);
}
int get_int(){
    char buffer [NBCHIFFREMAX+1]; scanf("%s", buffer);
    if (EchoActif) printf(">>echo %s\\n", buffer);
    return atoi(buffer);
}
// Instructions -----
// developpe -----
void traite_developpe(){
    Mot nom_specialite; get_id(nom_specialite);    int cout_horaire = get_int();
    printf(MSG_DEVELOPPE, nom_specialite, cout_horaire);
}
// interruption -----
void traite_interruption(){ printf(MSG_INTERRUPTION);
}
// Boucle principale -----
int main(int argc, char *argv[]) {
    if (argc>=2 && strcmp("echo",argv[1])==0 ) EchoActif = VRAI;
    Mot buffer;
    while (VRAI){
        get_id(buffer);
        if (strcmp(buffer,"developpe")==0) { traite_developpe(); continue; }
        if (strcmp(buffer,"interruption")==0) { traite_interruption(); break; }
        printf("!!! instruction inconnue >%s< !!!\\n", buffer);
    }
    return 0;
}
```