

Cartouche du document

Année : ING 1

Matière : Programmation C

Activité : Examen de rattrapage

Objectifs

Cet examen teste vos aptitudes en programmation en C

- à traduire en C des algorithmes
- à corriger les erreurs d'un source C
- à manipuler les pointeurs
- à utiliser la gestion bufferisée de fichiers en C

Tous les documents (électroniques et autres) sont autorisés.

La durée de l'examen est de 1h30.

Vos réponses à l'examen seront rédigées dans un unique fichier texte que vous envoyez par mail à florent.devin@eisti.fr pour les étudiants de Pau et à rc@eisti.fr pour les étudiants de Cergy. Dans les deux cas votre mail doit avoir pour sujet "Examen Ratt Prog C".

Sommaire des exercices

- 1 - Suites arithmétiques et géométriques (7pts)
- 2 - Les pointeurs (7pts)
- 3 - Quelques erreurs de compilation (4pts)
- 4 - Les fichiers (6pts)

Corps des exercices

1 - Suites arithmétiques et géométriques (7pts)

Enoncé :

On considère le type abstrait suivant :

```
TYPE ABSTRAIT TypeSuite
Concept : Ce type permet d'énumérer différents types de suites
Opération de base
Constructeur TypeSuite : suiteGeo() : TypeSuite
Opération de base
Constructeur TypeSuite : suiteAri() : TypeSuite
FIN DE TYPE ABSTRAIT
```

```
TYPE ABSTRAIT SuiteAriGeo
Concept : Ce type permet de modéliser une suite arithmétique ou géométrique
Opération de base
```

Examen d'algorithmique 1er semestre

```
Constructeur SuiteAriGeo : creerSuiteAriGeo(Reel u0, Reel coeff, TypeSuite ts) :
SuiteAriGeo
Opération de base
Observateur SuiteAriGeo : recU0() : Reel
Opération de base
Observateur SuiteAriGeo : recCoeff() : Reel
Opération de base
Observateur SuiteAriGeo : recTypeSuite() : TypeSuite

Début des axiomes
recU0(creerSuiteAriGeo(u0,coeff,ts)) = u0
recCoeff(creerSuiteAriGeo(u0,coeff,ts)) = coeff
recTypeSuite(creerSuiteAriGeo(u0,coeff,ts)) = ts
Fin des axiomes
FIN DE TYPE ABSTRAIT

Début d'opération d'extension
Transformateur Sortie : afficherTermesSuiteAriGeo(SuiteAriGeo sag,Entier n) :
Sortie
Transformé sortie
Objets Locaux
Reel u
Entier i
DEBUT
ecrireChaineNL(sortie,'Caractéristiques de la SuiteAriGeo :')
SI estEgal(recType(sag),suiteAri) ALORS
    DEBUT
    ecrireChaineNL(sortie,'Suite arithmetique')
    FIN
SINON
    DEBUT
    ecrireChaineNL(sortie,'Suite géométrique')
    FIN
ecrireChaine(sortie,'Premier terme = ')
ecrireReelNL(sortie,recU0(sag))
ecrireChaine(sortie,'Raison = ')
ecrireReelNL(sortie,recCoeff(sag))
SI estEgal(recTypeSuite(sag),suiteAri()) ALORS
    DEBUT
    u <-- recU0(sag)
    i <-- 1
    TANTQUE i <= n FAIRE
        DEBUT
        u <-- u + recCoeff(sag)
        ecrireChaine(sortie,'u(')
        ecrireEntier(sortie,i)
        ecrireChaine(sortie,') = ')
        ecrireChaineNL(sortie,u)
        i <-- i+1
        FIN
    FIN
SINON
    DEBUT
    u <-- recU0(sag)
    i <-- 1
    TANTQUE i <= n FAIRE
        DEBUT
        u <-- u * recCoeff(sag)
        ecrireChaine(sortie,'u(')
        ecrireEntier(sortie,i)
        ecrireChaine(sortie,') = ')
        ecrireReelNL(sortie,u)
        i <-- i+1
        FIN
    FIN
FIN
Fin d'opération d'extension
```

Question 1)

Enoncé de la question

Implémenter en C, le type abstrait SuiteAriGeo.

On précisera bien ce qui doit être mis dans les fichiers d'entête (.h) et les fichiers sources (.c).

2 - Les pointeurs (7pts)

Enoncé :

On désire implémenter en C le concept d'ensemble. On supposera pour simplifier que les éléments de l'ensemble sont des entiers. On ajoute un élément dans l'ensemble que s'il n'existe pas.

On pourra utiliser la structure suivante :

```
typedef struct
{
    int * pelements;
    int nbElements;
} SEnsemble;

typedef SEnsemble * Ensemble;
```

Question 1)

Enoncé de la question

Ecrire en C, la fonction qui crée un ensemble vide.

Question 2)

Enoncé de la question

Ecrire en C, la fonction qui permet d'ajouter un élément à un ensemble.

Question 3)

Enoncé de la question

Ecrire en C, la fonction qui permet de supprimer un élément d'ensemble.

3 - Quelques erreurs de compilation (4pts)

Enoncé :

Corriger les erreurs de ce squelette de source C (warning et syntax error). On utilisera la commande gcc avec l'option -c pour s'arrêter à la traduction du code : **gcc -Wall -c ???c** .

```
int main()
{
    int i;

    for(i = 0; i < 10; I++)
    {
        // to do
    }

    int k;
```

```
for(k = 0; i < 10; k++)  
{  
    / to do  
}  
}
```

4 - Les fichiers (6pts)

Enoncé :

Dans cet exercice, on manipule un fichier qui contient des nombres réels stockés en mode texte (formaté).

Question 1)

Enoncé de la question

Ecrire en C, une fonction qui reçoit le nom d'un tel fichier et qui stocke dans un fichier résultat le nombre de réels contenus dans le fichier ainsi que la somme de ces nombres. Le stockage se fera en mode binaire (mode non formaté). Le nom du fichier résultat sera le nom du fichier initial auquel on aura ajouté l'extension .res.