INF111 :
EC2 : Programmation en langage C
Session de décembre 2022
Durée : 1 heure

NB : documents, calculatrices et téléphones fermés. Ne rien écrire au crayon

QCM (5 pts) : reportez simplement le numéro de la question et la lettre correspondante à la bonne réponse (Exemple : 8 - a). Mauvaise réponse : -0.5 point

1- Soit la déclaration suivante: `int n=10, p=5, q=10, r = n << sizeof(p)` ;

Quelle est la valeur de r?

Réponses : a) 10 ; b) 160; c) 400; d) 5; e) Aucune réponse juste car l'instruction est erronée

2- Quelle sont les valeurs de n, p, q après exécution de ces deux instructions?

`int n=10, p=5, q=10, r ;`

`q = n < p ? --n : --p ;`

Réponses : a) q=4, n=9, p=4 ; b) q=9, n=9, p=4 ; c) q=0; n=10, p=5 ; d) q=5; n=9; p=4 ; e) q=6; n=11; p=6 ; f) Aucune réponse correcte parmi a, b, c et d car il y a erreur.

3- Quelle instruction initialise le tableau A à 2 dimensions (3x3) à la déclaration?

Réponses : a) `for (i=1; i<=5;i++) {A[i,j]=0; ++i};` b) `int A[3][3]={1,2,3,4,5,6,7,8,9};` c) `int A[3][3]={{1,2,3}, {4,5,6},{7,8}};` d) `int A[3][3]={1,2,3,...,9};` e) aucune réponse parmi a, b et c n'est juste

4- Considérer la déclaration suivante et choisir la bonne réponse :

`int n = 10 , p = 4 ; long q = 2 ; float x = 1.75 ;`

Réponses : a) Ces déclarations sont correctes ; b) Ces déclarations ne sont pas correctes car le type « long » n'existe pas en C ; c) Ces déclarations ne sont pas correctes car à cause du « = » utilisé

5- Avec la déclaration ci-dessous, quel est le type de l'expression `n + x`?

`int n = 10 , p = 4 ; long q = 2 ; float x = 1.75 ;`

Réponses : a) int ; b) long ; c) float ; d) long float ; e) Aucune réponse correcte parmi a, b, c et d.

Exercice : Le problème consiste à utiliser les instructions répétitives et les tableaux (2+1,5+1,5) pts

1. Ecrire un programme qui demande à l'utilisateur d'entrer un entier `n < 10` (le programme doit s'assurer) et affiche la table. Par exemple pour `n = 6`, on a :

```
9 x 1 + 2 = 11
9 x 12 + 3 = 111
9 x 123 + 4 = 1111
9 x 1234 + 5 = 11111
9 x 12345 + 6 = 111111
```

2. Ecrire un programme qui demande d'entrer un entier positif `n` et affiche le triangle rectangle des étoiles. Par exemple pour `n=5`, on a :

```
*
**
***
****
*****
```

3. écrire un programme qui demande à l'utilisateur d'entrer un entier `n` et affiche le triangle de Pascal d'ordre `n`

Université de Yaoundé 1
Licence 1 - EC2 : Programmation en langage C
Examen de fin de Semestre 1 - 2023
Durée : 1 heure

NB : document, ordinateur, téléphone et calculatrice fermés

Exercice 1 : Programmer en C

7 points

Un étudiant est décrit par son matricule (une chaîne de caractères) , son nom (une chaîne de caractères) et son âge (un entier). Le but est de traiter les données des étudiants d'une classe

1. Définir en C une structure de données pour représenter un étudiant, une autre de pour représenter une classe
2. Ecrire un programme qui demande à l'utilisateur d'entrer un entier positif n, puis d'entrer les données de n étudiants et afficher ces données dans l'ordre inverse.
3. Ecrire un programme qui affiche le nom de l'étudiant le plus jeune de la classe et le nom de l'étudiant le plus âgé.
4. Ecrire un programme qui affiche l'âge moyen de la classe
5. écrire un programme qui demande à l'utilisateur d'entrer un matricule et recherche si le matricule saisi fait partie de la classe ou pas.

Exercice 2 : traduire l'algorithme en programme C

3 points

NB : chaque faute (erreur) entraîne le retrait de 0,5 point

Algorithme recherche_sequentielle

Type Vecteur=tableau[1..100] de entier ;

Var T : Vecteur

Val, n, i : entier

Début

Ecrire ("entrer le nombre d'éléments") ;

Lire(n)

Ecrire ("entrer les éléments du tableau")

Pour i allant de 1 à n faire

 Lire (T[i])

finpour

Ecrire ("entrer l'élément à rechercher") ;

Lire(val) ;

i ← 1

tant que (val ≠ T[i] et i < n) faire

 i ← i + 1

 fintantque

si T[i] = val alors

 écrire(" l'élément se trouve en : ", i);

sinon

 écrire(" Elément non présent ");

fsi

Fin

Examen d'algorithmique février 2023

Problème (10 pts)

Les élections sénatoriales auront lieu le 12 mars prochain au Cameroun, selon un décret signé vendredi par le Président Paul Biya. Le vote se déroulera de 08H00 à 18H00 heure locale dans les chefs-lieux des départements du pays, précise le décret. Le Sénat, la chambre haute du parlement bicaméral du Cameroun, est composée de 100 sièges, dont 70 sénateurs seront élus au suffrage indirect par un collège constitué des conseillers municipaux et régionaux dans les dix régions du pays. Pour cette élection 13 partis politiques sont en lice. Dans chaque région, 7 sénateurs sont élus. Le Cameroun compte 58 départements et chaque département a son quota de sénateurs. La somme du nombre de sénateurs élus dans les départements d'une région devant être le chiffre 7 comme annoncé plus haut. Selon le nombre de place par département, l'élection a lieu au scrutin de liste à un seul tour. Les sièges sont attribués en fonction de l'ordre de présentation des candidats sur chaque liste.

Travail à faire :

1. Donner la structure de donnée sous forme d'un enregistre de la liste des candidats en compétition d'un parti politique à cette élection dans un département (1pt)
2. Donner sous forme d'un tableau la liste des candidatures reçues en compétition dans le département (1pt)
3. Donner la structure de donnée sous forme d'un enregistrement le choix du votant sachant que ce dernier est transparent lors des décomptes des votes (1pt)
4. Sachant que le nombre de votants n'excède pas 1000 par département, donner la structure de donnée sous forme d'un tableau des votes exprimés (1pt)
5. Définir l'algorithme *DECOMPTE* qui procède au décompte des votes dès la fin du vote (1,5pt)
6. Définir l'algorithme *CLASSER* qui classe les partis politique par ordre de votes reçus (1pt)
7. Définir l'algorithme *CALCUL-POURCENTAGE* qui calcule le pourcentage des votes de chaque parti politique (1,5pt)
8. Définir l'algorithme *PROCLAME* qui proclame les résultats du vote (1pt)
9. Définir l'algorithme *ELUS* qui liste les noms des sénateurs élus dans le département. (1pt)

Programmation en C
Fiche de TD&TP N°1

Exercice 1 : Soit n un entier non signé déclaré par `unsigned int n`. Donnez la ou les instruction(s) permettant de

1. mettre tous les bits de n à 0
2. mettre tous les bits de n à 1
3. mettre tous les bits de n à 0 sauf le bit numéro i qui doit être à 1
4. mettre tous les bits de n à 1 sauf le bit numéro i qui doit être à 0
5. forcer le bit numéro i de n à passer à 1
6. forcer le bit numéro i de n à passer à 0
7. inverser le bit numéro i
8. mettre tous les bits d'indice strictement inférieur à i à 1, et tous les autres bits à 0
9. mettre tous les bits d'indice strictement inférieur à i à 0, et tous les autres bits à 1
10. forcer tous les bits de n d'indice inférieur ou égal à i de passer à 1
11. forcer tous les bits de n d'indice inférieur ou égal à i à passer à 0
12. tester si le bit d'indice i est égal à 0
13. tester si le bit d'indice i est égal à 1
14. calculer le nombre de bits à 1 dans n
15. inverser l'ordre des bits de n (le bit de poids fort doit devenir bit de poids faible, le bit de poids faible doit devenir bit de poids fort, etc...)
16. échanger les bits d'indice pair avec les bits d'indice impairs (on supposera que n est un nombre de 32 bits).
17. inverser l'ordre des octets dans le nombre n , que l'on supposera être un nombre de 32 bits (passage de big endian à little endian ou réciproquement)

Exercice 2: Ecrivez une fonction qui multiplie un nombre a par un nombre b en n'utilisant que des additions, des décalages et des opérations bit à bit.

Exercice 3: Écrire un programme C qui lit un entier R au clavier représentant le rayon d'un cercle, calcule et affiche le périmètre de ce cercle ainsi que sa surface.

Exercice 4: Écrire un programme C qui lit un réel M au clavier représentant une moyenne et affiche la mention correspondant à cette moyenne.

Exemple : si M est dans l'intervalle $[10-12[$ alors mention passable

Exercice 5 : Écrire un programme C qui lit un entier n (s'assure que n est strictement positif) et calcul le factoriel de cet entier dans les deux cas suivants:

a- En utilisant uniquement la boucle `for`.

b- En utilisant uniquement la boucle `while`.

Exercice 6 : Écrire un programme C qui lit deux entiers a et b au clavier, calcule et affiche pgcd de ces entiers.

Exercice 7:

```
do {
    // instructions
}while(condition)

while(condition){
    // instructions }
}
```

a- Quelle est la différence entre ces deux boucles ?

b- Quand est-il convenable d'utiliser le «do-while»? Donnez un exemple.

Exercice 8 : Ecrire un programme C qui lise 10 entiers qu au clavier et les affiche dans l'ordre inverse de lecture. Utilisez un tableau pour le faire.

Exercice 9 : Ecrire un programme C qui initialise deux tableaux d'entiers **t1** et **t2** en lisant des entiers au clavier, calcul le produit **t1x2** et stocke le résultat dans **t3**.

$$\text{Formule : } t3[i] = t1[i] \times t2[i]$$

Exercice 10 : Ecrire un programme C qui initialise deux matrices carrées A et B d'ordre n (lu avant la lecture des matrices) en lisant des valeurs au clavier, calcul et affiche la somme $C = A+B$.

$$\text{Formule : } C[i][j] = A[i][j] + B[i][j]$$

Exercice 11 : Ecrire un programme C qui initialise deux matrices carrées A et B d'ordre n (lu avant la lecture des matrices) en lisant des valeurs au clavier, calcul et affiche le produit $C = A \times B$.

Exercice 12: Considérer l'algorithme suivant qui a pour objectif de lire un entier $n > 1$ et afficher un message indiquant si n est un nombre premier.

```
Algorithme Premier
var n, d, q, r : entier
début
    répéter lire (n) jusqu'à (n>1)
    d ← 2
    q ← n div d
    r ← n - q * d
    tant que (r > 0) et (d < n-1) faire
    début
        si d = 2 alors d ← d + 1
        sinon d ← d + 2
        q ← n div d
        r ← n - q * d
    fin
    si (r = 0) alors écrire (n, "n'est pas premier")
    sinon écrire (n, "est premier")
fin
```

- Ecrire le programme C correspondant à cet algorithme
- Ce programme (ou algorithme) ne donne pas toujours les réponses souhaitées. Justifier cette assertion.
- Donner sans modifier la boucle « tant que ... », une version de l'algorithme Premier (ou du programme) qui donne toujours des bonnes réponses.
- Considérer la bonne version de votre algorithme Premier. La condition de fin $(r > 0)$ et $(d < n-1)$ est très naïve. Pourquoi ?
- Proposer une nouvelle version de l'algorithme Premier (et du programme) qui utilise une meilleure condition de fin et dire pourquoi cette nouvelle version est plus performante.

Exercice 13

On veut déterminer la liste des nombres premiers inférieurs ou égaux à 1000. Pour ce faire, on utilise un tableau Table dont l'élément numéro i est initialisé à i .

- Proposer un sous-programme qui initialise tous les éléments du tableau Table.
- Ecrire un sous-programme qui pour un entier j passé en paramètre, (j inférieur ou égal à 1000) met à zéro tous les éléments du tableau qui sont des multiples de j différents de j .
- Dans la fonction main () appelez le sous-programme de la question précédente en lui passant comme paramètre 2.
- Le plus petit nombre supérieur strictement à Table[2] correspondant à un élément du tableau Table qui n'a pas été mis à zéro est un nombre premier. On peut généraliser en disant que si Table[i] est différent de 0, et que tous les éléments multiples de j dans le tableau Table sont à 0 alors le plus petit nombre supérieur strictement à Table[i] correspondant à un élément du tableau Table qui n'a pas été mis à zéro est un nombre premier ; utiliser cette idée, les idées des questions a), b) et c) pour développer un algorithme qui détermine la liste des nombres premiers inférieurs ou égaux à 1000.

Exercice 14

On suppose qu'un carré magique est une matrice carrée dont la somme des éléments de n'importe quelle colonne ou de n'importe quelle ligne est égale à la même constante CTE.

- Ecrire une fonction qui prend en paramètre une matrice M et un entier i puis calcule et retourne la somme des éléments d'une ligne i
- Ecrire une fonction qui prend en paramètre une matrice M et un entier j puis calcule et retourne la somme des éléments de la colonne j
- Ecrire un programme principal qui utilise les deux fonctions précédentes pour vérifier qu'une matrice carrée d'ordre n entrée par l'utilisateur est un carré magique.

Exercice 15:

Considérer l'algorithme de suivant :

Algorithme TRIECHANGE

Début

 Répéter

 Inversion $\leftarrow$ vrai

 Pour $i=1$ à $N-1$ faire

 Si $A[i] > A[i+1]$ alors Début

 Aux $\leftarrow A[i+1]$

$A[i+1] \leftarrow A[i]$

$A[i] \leftarrow$ Aux

 Inversion $\leftarrow$ faux

 Fin

 Jusqu'à inversion=faux

 Fin

- Ecrire le programme C correspondant à cet algorithme
- on suppose que N est égal à 8 et que le vecteur A contient dans l'ordre des indices croissants les valeurs réelles suivantes : 10, 2, 11, 20, 12, 8, 9, 10. Modifier le programme écrit à la question pour afficher les valeurs des différents objets manipulés dans la programme à chaque passage dans la boucle « répéter... jusqu'à »

Exercice 16 :

Considérer un vecteur Vec de N nombres réels. On veut vérifier si une valeur Val donnée fait partie des N nombres réels que contient le vecteur Vec .

- On suppose que le vecteur est trié dans l'ordre croissant. Donner l'algorithme de recherche dichotomique.
- Ecrire le programme correspondant à l'algorithme écrit à la question a).
- On suppose que le vecteur est trié dans l'ordre décroissant. Modifier l'algorithme et le programme écrits précédemment pour rechercher Val dans Vec .
- On suppose que le vecteur n'est pas trié. Peut-on encore appliquer l'algorithme de recherche dichotomique ? Pourquoi ?

Exercice 17

Considérer l'algorithme de recherche suivant :

```
Algorithme RechercheSentinelle
Début
    //Lecture des entrées
    Vec[N+1] ← Val ;
    i ← 1
    Tant que (Vec[i]≠Val) faire
        i ← i+1
    Ftantque
    Si (i>N) alors Trouve ← faux
    Sinon
        Trouve ← vrai
        Pos ← i
    Fsi
    //Affichage des résultats
fin
```

- Écrire des instructions pour faire la lecture des entrées et pour afficher les résultats
- Écrire le programme C équivalent à cet algorithme
- Modifier le programme pour qu'il affiche à chaque itération de la boucle « tant que... » les valeurs de i , $Vec[i]$ et de Val . Qu'affiche ce programme ?
- Modifier l'algorithme en surprenant l'instruction $Vec[N+1] \leftarrow Val$ et en arrangeant la suite pour que l'algorithme fasse exactement la recherche. Modifier aussi les programmes des questions b) et c) pour prendre en compte ces modifications.