

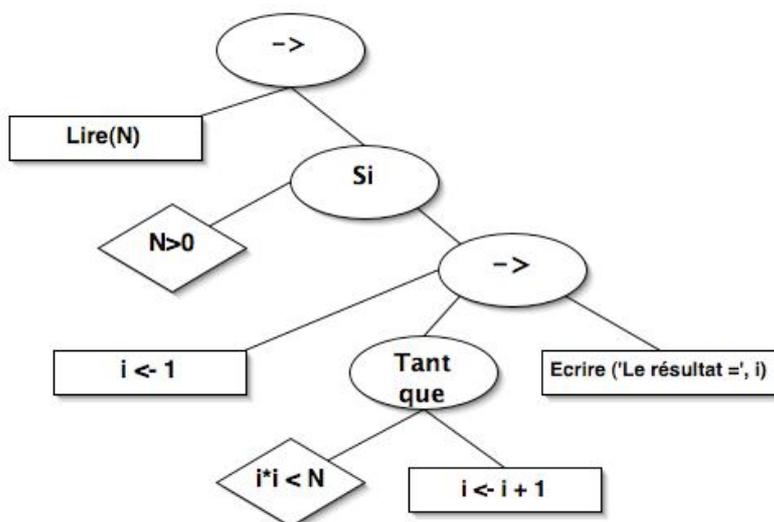
## RAN Informatique - FA14

NOM :	Prénom :
Filière :	

Seuls les photocopiés distribués en cours sont autorisés. Si vous manquez d'espace pour écrire utilisez la feuille d'examen fournie.

### Exercice 1

Soit l'arbre programmatique suivant (les variables sont des entiers):



Q 1.1) Que fait cet algorithme?

Q 1.2) Traduire cet algorithme en langage C.

## Exercice 2

On considère le code en C, suivant :

```
#include <stdlib.h>
#include <stdio.h>

#define TRUE 1
#define FALSE 0

// fonction 1
int myster1(int m)
{
    int i ;
    for(i = 2; i < m; i = i + 1)
        if ((m % i) == 0)
            return FALSE;
    return TRUE;
}

// fonction 2
int myster2(int n)
{
    while(! myster1(n))
        n = n + 1;
    return n;
}

// La fonction principale
int main()
{
    int k;
    printf ("Entrez une valeur pour la variable k : ");
    scanf ("%d", &k);
    printf ("%d", myster2(k));
    return 0;
}
```

Q 2.1) : Que fait ce programme ?

Q 2.2) Quel est le résultat qu'il affiche si l'on saisie la valeur de 12 ?

Q 2.3) Quel serait le résultat si l'on saisie la valeur de 21 ?

### Exercice 3

Ecrire un programme principal **int main()** qui lit une ligne de texte (ne dépassant pas 200 caractères) la mémorise dans une variable **Line** et appelle les deux fonctions suivantes:

Q 3.1) Une fonction (ou procédure) **inverse** qui inverse l'ordre des caractères dans **Line** et affiche le résultat:

Exemple :

Ligne saisie : *voici une petite phrase !*

Résultat d'affichage : *! esarhp etitep enu iciov*

Q 3.2) Une fonction (ou procédure) **enleve** qui enlève toutes les occurrences du caractère 'e', dans **Line**, en tassant les éléments restants et affiche le résultat. Les modifications se feront dans la même variable **Line**.

Exemple :

Ligne saisie : *Cette ligne contient quelques lettres e.*

Résultat d'affichage : *Ctt lign contint qulqus lttrs .*

```
#include <stdlib.h>
#include <stdio.h>

// Q 3.1) fonction 1 à compléter
```

```
// Q 3.2) fonction 2 à compléter
```

```
// La fonction principale
int main()
{
    char Line[200];
    printf ("Entrez une phrase qui ne dépasse pas 200 caractères : ");
    scanf("%s", Line);
    // Appel de la fonction 1 à compléter

    // Appel de la fonction 1 à compléter

    return 0;
}
```

#### Exercice 4

Un étudiant sera caractérisé par :

- son numéro d'étudiant,
- son nom,
- son prénom,
- sa date de naissance (qui sera décomposée en jour mois et année de type entier)
- un tableau contenant les notes de 5 modules

Numéro d'étudiant	1060333
Nom d'étudiant	Leroy
Prénom d'étudiant	Lucy
Date de naissance	20
-jour	05
-moi	1988
-année	
Tableau de notes	12   15   17   14   8

Q 4.1) Écrire en C les structures de données nécessaires à la définition d'un étudiant (***typedef struct Etudiant***) avec toutes les caractéristiques citées ci-dessus.

Note : La date de naissance doit être stockée dans une structure à part.

Q 4.2) Écrire une fonction en respectant le prototype suivant ***struct Etudiant saisieEtudiant( )*** qui invite à la saisie d'un étudiant au clavier.

Q 4.3) Ecrire une procédure de prototype ***void afficheEtudiant(Etudiant e)***, qui affiche les caractéristiques de l'étudiant passé en paramètre.

Q 4.4) Ecrire une fonction de prototype ***int egales(Etudiant e1, Etudiant e2)*** qui compare deux étudiants ***e1*** et ***e2*** par leurs notes moyennes et retourne 0 s'ils ont des moyennes différentes et 1 sinon.

Q 4.5) Ecrire un programme principal ***int main()*** qui permet d'appeler toutes ces fonctions et procédures, en n'oubliant pas de déclarer de variable nécessaires.

### **Exercice 5**

Ecrire un programme qui remplace, dans un fichier contenant un texte, les retours à la ligne par des espaces. Si plusieurs retours à la ligne se suivent, seulement le premier sera remplacé. Les noms des fichiers source et destination sont entrés au clavier.

**Bonne chance.**