NOM Lellouche

Prénom Léo

Promo 2020

Date 17/05/16

LELLOUCHE Léo
L1 - 2015

14,5

MATIÈRE Algo

Exercice A :

Algorithme : Factoriel (n : entier naturel) : entier naturel

Valeur de retour : entier (naturel), factoriel du nombre passé en paramètre

Paramètre : n, entier naturel, nombre dont on souhaite connaître la factoriel.

```
DEBUT :
|  Si (n = 1)
|  |  Retourner (1)
|  Sinon
|  |  Retourner (Factoriel (n - 1) x n)
|  Fin Si
FIN
```

A

On sait que $n! = n \times (n-1)!$

On sait aussi que $1! = 1$

On va donc chercher à faire une recursive où il faudra calculer $n \times (n-1) \times (n-2) \times \ldots \times 2 \times (1!)$

Et de remplacer $1!$ par 1.

Exercice B :

Structure : Fiche : Nom, chaine de caractères
Prenom, chaine de caractères
Age, entier naturel

Algorithme : Fiche () : Fiche

B

Valeur de retour : Structure fiche

Variable locale : tmp, grande chaine de caractères servant à stocker temporairement le nom et le prenom
Infos, structure fiche

DEBUT : Infos ← Réserver mémoire (pointeur)
Afficher "Quel est votre nom ?"
tmp ← Saisir
Infos.Nom ← Reserver (strlen(tmp)) (voir plus tard)
Strcpy (Infos.Nom, tmp)
Afficher "Quel est votre prénom ?"
tmp ← Saisir
Infos.Prenom ← Reserver (strlen(tmp))
Strcpy (Infos.Prenom, tmp)
Afficher "Quel age avez vous ?"
Infos.Age ← Saisir
FIN Retourner (Infos)

Algorithme : Strlen (str : chaine de caracteres) : entier

Valeur de retour, entier naturel, taille de la chaine

Parametre : str, chaine de caractère

Variable locale : i, entier, taille de la chaine

```
DEBUT
    i ← 0
    Tant que (str[i] ≠ '\0')
        i ← i+1
    Fin Tant Que
    Retourner (i+1)
FIN
```

Algorithme : Strcpy (dst, src : chaines de caractere)

Parametres : dst, chaine de caracteres, destination
src, chaine de caracteres, source

Variable locale : i, entier, curseur

```
DEBUT :
    i ← 0
    Tant que (src[i] ≠ '\0')
        dst[i] = src[i]
    Fin Tant que
    dst[i] = '\0'
FIN
```

Texte de justification à la toute fin de ma copie

Exercice C :

Algorithme : Initialise2D(a : tableau 2D de taille n x n)

Paramètre : a : tableau 2D (d'entiers) de taille n x n

Variable locale : i, entier statique, initialisé à Taille-tableau(a) - 1

nb, entier statique, initialisé à 0.

DEBUT :

| Remplir2D(a[i], nb, Taille-tableau(a) - 1)
| i <- i - 1
| nb <- nb + 1
| Si (i < 0)
| | Retourner
| Sinon
| | Initialise2D(a)
| Fin Si
FIN

quelle est sa valeur ?

C

Algorithme : Remplir2D(a : Tableau 1D d'entiers, nb : entier, taille : entier)

Paramètres : a : Chaine que l'on va remplir
nb : nombre que l'on va placer (de droite à gauche)
taille : taille du tableau

NOM Lellouche

Prénom Léo

Promo 2020

Date 17/03/16

MATIÈRE Algo

Exercice 6 suite :

```
DEBUT :
    a[size] <-- nb
    size <-- size - 1
    Si (size < 0)
        Retourner
    Sinon
        Remplir2D(a, nb+1, size)
    FinSi
FIN
```

Il s'agit ici de remplir ligne à ligne en partant du bas.

Le nombre tout en bas à droite sera 0. Il faut alors remplir la ligne de droite à gauche en incrementant à chaque fois le nombre que l'on inscrit

A chaque fois qu'on monte une ligne au dessus, on incremente le nombre que l'on met à droite

On s'arrete lorsque l'on a tout rempli.

Exercice I :

Structure : Complexe : Re, nombre reel
Im, nombre reel

Algorithme : Limite (C, zn : Complexe) : Complexe

Constante : e, nombre réel

Parametres : C, structure complexe
zn, structure complexe, c'est zn n

Variable locale : suivant, structure complexe : zn n+1

DEBUT :

    Suivant.Re ← (zn.Re + zn.Re) − (zn.Im × zn.Im) + C.Re

    Suivant.Im ← (2 × zn.Re × zn.Im) + C.Im

    Si (Mod (suivant, zn) < e)

        Retourner (Suivant)

    Sinon

        Retourner (Limite (C, suivant))

    Fin Si

FIN

Algorithme : Mod(z_1, z_0 : Complexes)

Paramètres : z_0, complexe, z_n
z_1, complexe, z_{n+1}

Variables locale : tmp, complexe, différence de coordonnée des 2 points
Module, nombre reel, module

DEBUT :

tmp.Re ← z_1.Re − z_0.Re

tmp.Im ← z_1.Im − z_0.Im

Module ← (tmp.Re × tmp.Re) + (tmp.Im × tmp.Im)

Module ← sqrt(Module)

Retourner(Module)

FIN

On écrit la valeur de z_{n+1} dans la variable suivant.
On calcul le module et on le compare à epilon
Si on est inférieur, on a la limite. Sinon, on recommence.

Exercice B explication

On demande à l'utilisateur chaque info.
Pour le prenom et le nom, on stock ça ~~dans une liste~~ dans une grande chaine statique, puis on copie dans une chaine où la mémoire aura correctement été aloué en fonction de la taille du nom et prenom.

D/