Chapitre 2:

La récursivité

Activité

Ecrire un algorithme d'une fonction Fact permettant de calculer la factorielle d'un entier n supérieur ou égal à 0.

On rappelle que la factorielle d'une entier positif n notée en mathématiques n! est calculée comme suit :

$$\begin{cases} 0! = 1 \\ n! = n * (n-1) * (n-2) * \dots 3 * 2 * 1 \text{ (pour n>0)} \end{cases}$$

Exemple: 5! = 5*4*3*2*1 = 120

forction Fact (n:entier): entier

f=1

Pour i de la néaire

linfour retourner f

Solution intérative

La factorielle d'un entier peut être calculée d'une autre façon.

Exemple: Pour n = 5

Fact (5) =
$$5*$$
 $4*3*2*1$ Fact (4)

n! = n * (n-1)!

A partir de cet exemple, proposer une autre

formule pour calculer la factorielle de n.

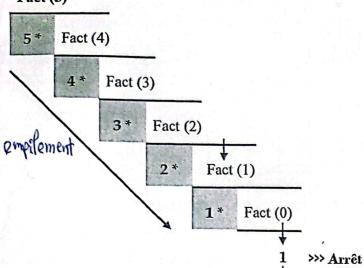
Formule 2:

Enseignante: ICHRAF MABROUK

Explication:

Fact (5)





5 *	Fact (4)	= 120
4 *	Fact (3)	= 24
3 *	Fact (2)	= 6
2 *	Fact (1) -	= 2

Fact (0)



Solution récursive :

fauction fact (n.	entrer leadons
0	
Debut	

		Λ	
 .si.m.	J.Q.	alo	٠
	4		λ

Stron	
	A STATE OF THE STA
retourner.	ny fact (not)
0	, , 0

	rotour .	la vi	Pach
	retourne		9001.4.
	i		ģ
Oin	S.i		
N. 4.r		••••••	

Définitions

- La récursivité est une méthode algorithmique qui consiste à appeler un module dans son propre corps.
- On appelle récursive toute fonction ou procédure qui s'appelle elle-même.
- Un module récursif s'appelle lui-même avec de nouvelles valeurs de paramètres effectifs dans chaque appel jusqu'à atteindre une condition d'arrêt.
- Un traitement récursif doit comporter :
 - Une (ou plusieurs) cas d'arrêt de tout appel récursif.
 - Un cas général (ou plusieurs cas généraux) dans lequel un ou plusieurs appels seront effectués (en changeant les paramètres)

Applications 1) Calcul de la puissance d'un entier Ecrire un algorithme d'un module permettant de calculer la puissance n^e d'un entier $n \ge 0$. 2) Calcul du pgcd de deux entiers (principe des différences) Ecrire un algorithme d'un module permettant de calculer le PGCD de deux entiers strictement positifs en utilisant le principe des différences. 3) Somme des chiffres d'un entier Ecrire un algorithme d'un module permettant de calculer la somme des chiffres d'un entier positif. 4) Somme des chiffres d'une chaîne numérique : Ecrire un algorithme d'un module permettant de calculer la somme des chiffres d'une chaîne numérique. 5) Vérification si une chaîne est un palindreme: Ecrire un algorithme d'un module permettant de vérifier si une chaîne est palindrome ou non. Conction pacalarbientien :entre Debu Debut sia=balors 81 e = 0 alor retourner a etourner SINON yetourner ax puiss a e-1 BINOT insI retourner onction somme (n:enter) entier Debut SI n=0 alors retourner O Sinon retourner mod 10 + somme In in si

4) fonction someth (ch. chaine): entrer petet si long (ch) = 0 alors retourner O sinon retourner Naleur (chip) + some on sous chaine(ch, 1, fong (ch finsi finsi si long (ch) = 0 ou long (ch) = 1 alors retourner Wai sinon si chia = chila = chilong (ch) - 1 alors retourner faus. 9; non retourner paland (sous chaine(ch, 1, fong (ch) - 1) finsi finsi finsi finsi finsi finsi finsi				
Debut si long (ch) = 0 alors retourner valeur(ch(o)) + som en (sous chaine(ch, 1, 1, long (ch) finsi fin S) fanction palan d(ch:chaine) : booleen Debut si long (ch) - 0 ou long (ch) = 1 alors retourner Urai si en (o) = ch [long (ch) - 1] erlors retourner faux si er aurner paland (sous chaine(ch, 1, long (ch) - 1) lins lins lins lins lins lins				
Debet Si long (ch) = 0 alors Yetourner valeur(ch(a)) + som on sous-chaine(ch, 1, long (ch finsi fin Conction pallonal ch: chaine): booleen Debut Si long (ch) = 0 ou long (ch) = 1 alors Yetourner Wai Sinon Si eh (a) = ch [long (ch) - 1) alors Yetourner faus Sinon Yetourner pland sous chaine(ch, 1, long (ch) - 1) Linsi Linsi	10%	5)	4)	
Debut Si long (ch) = 0 alors Yetourner O Sinon Tetourner valeur(ch[o]) + Som ch (sous chaine(ch, 1, long (ch finsi In Action pallon al ch: chaine): booleen Debut Si long (ch) = 0 ou long (ch) = 1 alors Yetourner Vai Sinon Si eh [o] = ch [long (ch) - 1] salors Tetourner faus Sinon Yetourner paland (sous chaine(ch, 1, long (ch) - 1) Pinsi Pinsi Pinsi		60	bom	
long (ch) = v alors vetourner o non retourner valeur (ch(o)) + som en (sous chaine (ch, 1, florey (ch si on pellon of ch: chaine y: booleen ut i long (ch) = o ou long (ch) = 1 alors xetourner Vrai man si : ch [o] = ch [long (ch) - 1] relors retourner faux Sinon retourner edand (sous chaine ch, 1, long (ch) 1) linsi linsi linsi	fin	nct. Deb	Debu Si	
ng(ch)=0 alors Pourner valeur(ch[o]) + som en sous-chaine(ch, 1 fong (ch) palan d (ch: chaine): booleen mg(ch) = 0 ou long(ch) = 1 alors courner vai -eh [o] = ch [long(ch) - 1] alors retourner faus non retourner paland sous chaine ch 1 fong(ch) 1)	Pins Dins	ion i fi ve vov	tel re	
ch)=0 alors ner valeur(ch(0))+ som en (sous-chaine(ch, 1, long (ch and (ch: chaine); booleen ch)=0 ou long (ch)=1 alors ner yrai 160) = ch [long (ch)-1] alors tourner palared sous chaine(ch 1, long (ch) 1)	n S i	ong l coux	ng (Fou	
= valeur(ch(o)), som in (sous, chaine(ch, 1, long (ch) al ch: chaine); booleen - o ou long (ch) = 1 alors Vai 3 = ch [long (ch) - 1] alors ener faux		ch) nex	ch) rner	
alors leux(ch(0)) + som ch(sous chaine(ch, 1 flores (ch chaine); booleen ou long(ch) = 1 alors ch [long(ch) - 1] alors Chause	17.5	-0 - Vi	- o	
ors v(ch(o)) + som ch (sous chaine(ch, 1, hong(ch) laine); booleen long(ch) = 1 alors aus.		ou cai	ol	
h(0)) + som en (sous-chaine(ch, 1-hong(ch)); booleen g(ch) = 1 alors na(ch) - 1) alors		lon all	brs	
)-+ som en (sous-chaine(ch, 1, long (ch) coleen ch)-1 alors		alc'		
som ch (sous chaine(ch, A flong (ch) A alors 1) alors	4	h) =		
Lors	OA	Λ. _Λ.		
04.5	(A)	\ cel	A. ch	
		045	\ sou	
		, 1, 1, 1	s_ch	
		Pone	ıcine	
Arlongton		(c)	(ch	
Congleh		-1)	, 1,	
(ch			ony	
			ljch.	