

TP : Les boucles

Charles Yaiche avec l'aide de Abdul Farouk
charles.yaiche.lf@gmail.com

Architecture de rendu



Les fonctions seront écrites dans dans le fichier main.c en dessous de la fonction main.



Exercice 1

Description	
Écrire une procédure pour afficher tous les nombres naturels de 1 à n en utilisant la boucle for	
Prototype abstrait	Prototype de fonction
nombre : Entier -> Vide	void nombre(int a);
Exemples	Résultat
nombre (5) ;	//affiche : "1,2,3,4,5"
Conseils	



Exercice 2

Description	
Écrire une procédure pour afficher tous les nombres naturels en sens inverse de n à 1 en utilisant la boucle while.	
Prototype abstrait	Prototype de fonction
inverse : Entier -> Vide	Vide inverse (int a);
Exemples	Résultat
<code>inverse(5);</code>	//affiche : "5,4,3,2,1"
Conseils	



Exercice 3

Description	
Écrire une procédure pour afficher tous les nombres pairs de 1 à n en utilisant la boucle for ou while.	
Prototype abstrait	Prototype de fonction
pairs : Entier -> Vide	void pairs (int a);
Exemples	Résultat
<pre>pairs(11);</pre>	<pre>//affiche : "les nombres pairs de 1 à 11 sont : 2 4 6 8 10 "</pre>
Conseils	



Exercice 4

Description	
Écrire une procédure pour afficher tous les nombres impairs de 1 à n en utilisant la boucle for et while.	
Prototype abstrait	Prototype de fonction
impairs : Entier -> Vide	void impairs (int a);
Exemples	Résultat
<code>impairs(11);</code>	//affiche : "les nombres impairs de 1 à 11 sont : 1 3 5 7 9 11 "
Conseils	



Exercice 5

Description	
Écrire une procédure pour saisir un nombre et recherchez le premier et le dernier chiffre d'un nombre en utilisant une boucle.	
Prototype abstrait	Prototype de fonction
trouver : Entier -> Vide	void trouver(int n);
Exemples	Résultat
<code>trouver(1980);</code>	//affiche : "Premier chiffre=1 Dernier chiffre= 0"
Conseils	



Exercice 6

Description	
Écrire une procédure qui affiche les table de multiplication d'un nombre n et afficher la table de multiplication allant de start, à end en utilisant "for"	
Prototype abstrait	Prototype de fonction
multi : Entier x Entier x Entier -> Entier	int multi (int n, int start, int end);
Exemples	Résultat
<code>multi(9, 5, 11)</code>	//affiche : 9 x 5 = 45 9 x 6 = 54 9 x 7 = 63 9 x 8 = 72 9 x 9 = 81 9 x 10 = 90 9x 11 = 99
Conseils	



Exercice 7

Description	
Écrire une procédure n pour entrer un nombre et afficher tous les facteurs du nombre donné en utilisant la boucle for.	
Prototype abstrait	Prototype de fonction
facteur : Entier -> Vide	void facteur (int i);
Exemples	Résultat
facteur (20) ;	//affiche : "Les facteurs de 20 sont : 1 2 4 5 10 20"
Conseils	
Le facteur de tout nombre est un nombre entier qui divise exactement le nombre en un nombre entier sans laisser de reste. Par exemple, 2 est un facteur de 6 car 2 divise 6 exactement sans laisser de reste.	



Exercice 8

Description	
<p>Écrire une fonction pour déterminer si le nombre est un nombre premier ou non en utilisant la boucle for.</p> <p>renvoi 0 si premier</p> <p>1 sinon.</p>	
Prototype abstrait	Prototype de fonction
prime : Entier -> Entier	int prime (int a);
Exemples	Résultat
<pre>int a = prime(11); printf("%d", a);</pre>	<pre>//affiche 0</pre>
Conseils	
<p>Un nombre premier est un entier naturel qui admet exactement deux diviseurs distincts entiers et positifs. Ces deux diviseurs sont 1 et le nombre considéré</p>	



Exercice 9

Description	
Écrire une fonction pour compter le nombre de chiffres dans l'entier donné en utilisant une boucle.	
Prototype abstrait	Prototype de fonction
chiffres : Entier -> Entier	int chiffres(int num);
Exemples	Résultat
<pre>int a = chiffres(2020); printf("%d");</pre>	<pre>//affiche : 4</pre>
Conseils	



Exercice 10

Description	
Écrire une procédure pour entrer un nombre et vérifiez si le nombre est parfait ou non.	
Prototype abstrait	Prototype de fonction
positif : Entier -> Vide	void positif (int i);
Exemples	Résultat
<code>positif(6);</code>	//affiche : "6 est un nombre parfait"
Conseils	
<p>Un nombre parfait est un entier positif qui est égal à la somme de ses diviseurs positifs appropriés.</p> <p>Par exemple: 6 est le premier nombre parfait</p> <p>Les diviseurs appropriés de 6 sont 1, 2, 3.</p> <p>Somme de ses diviseurs stricts = $1 + 2 + 3 = 6$.</p> <p>Par conséquent, 6 est un nombre parfait.</p>	



Exercice 11

Description	
Écrire une procédure pour afficher la série de Fibonacci jusqu'à n termes en utilisant une boucle.	
Prototype abstrait	Prototype de fonction
Fibonacci : Entier -> Vide	void Fibonacci (int a);
Exemples	Résultat
Fibonacci(5);	//affiche : "Termes de Fibonacci : 0 - 1 - 1 - 2 - 3 - 5 - 8 "
Conseils	
google fibonacci	