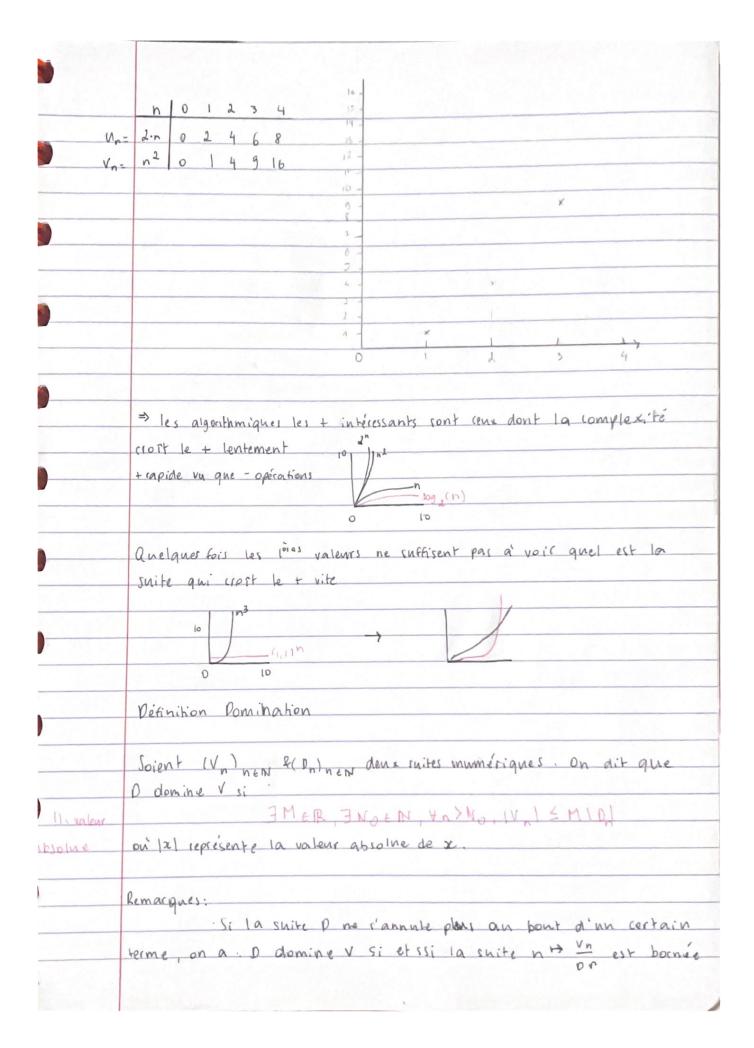
Analyse d'algorithmes					
Résond-t-il le problème?	Résond-t-il le publème?				
	go: pour rimporte quelle entrée,				
	" sorties conformes au problème				
Problème: proposio bgique	seur les E/S de l'algo				
Est-il performant?					
· Complexité en tem	ps: combien d'opé élem. effectuer				
	: place en mémoile?				
· (omplexité en esporce: place en mémoile).					
· ·	nne la classe de domina à laque				
appartient cette snite: 0 () on 0 ()					
ex: tii de tableau					
[1,2,3] -> 7 opé					
[32,1] > 18) opé ube varie celon entre. $[1,7,4,28] \rightarrow 108$					
mesure de la faille d'entrée pour tronver la valeur					
max des opé utilisées					
taille nhe ope					
3 18 ('pice" cas)					
4 27	pent se caractériser en classe				
5 100	d'équivalence des suites				
1 46 n2					
	Complexité				
1) Dans le pise des cos → très loura, l persimiste					
2) En moyenne: moy du noe d'opé/place occupée selon une					

distribus approba sur les objets de taille n. souvent de la distribus uniforme - méthoda préférée ex: tableanx analyse int tous les tableanx de toulle lour (£13) on étable le noe apré pour chaque (£2) touble et an en éait la mayenne (£3) (comportement quand nor 2 n-): (tn) ex: algo tri de tableanx de noe entiers ordre forcée: un tablean d'entiers E, un ordre 0 sur les entiers ordre soite: un tablean d'entiers S. Propriète: le tablean J soit trié selon l'ordre 0 le tablean J peur être obtenu en effectuant des échanges de valeurs entre les cases du tablean E. Mesure: on effectue der analyses de complexités en fonction du noe de cases du tablean E. (coissant / Tri à bulles: 1						
Aithon uniforme méthodo préférée ex: hableanx (t 1) (t 1) (t 2) (t 2) (t 2) (i 2) (i 3) (comportement quand not 2 n-1 (comportement quand not 2 n-1) (in) ex: q1go tri de tableanx de noe entiers ordre fincée: un tablean d'entiers E, un ordre 0 sur les entiers Propriétée: le tableau 1 soit trié selan l'ordre 0 Mesince: on effectua des analyses de complexités en fonction du noe de cases du tableau E. (coissant / Tri à bulles: 11 parroun 7 3: 10, 2:1:3:1.4 3:7:1:3:1:4:10 ox 3:7:10:1:3:1.4 3:7:1:3:1:4:7:10 ennange		la la la la la no souvent				
ex: hableanx analyse sur tous les tableanx de touille 10 (td) (td) (td) (ill et on en éait la mayenne (ex) algo tri de tableanx de nhe entiers ex: alg		distribue de proba sur les objets de taille				
(th) on Endic le Noe opé pour chaque (th) bille et on en rait la moyenne [t3] (comportement quand not 2 n-) (tn) ex: 919.0 tri de tableaux de noe entiers propriétée: un tableau d'entiers & un oidre 0 sur les entiers Propriétée: le tableau d'entiers & les consecution des échanges de valeurs entre les cases du tableaux E. Mesure: on effectue des analyses de complexités en fonction du noe de cares du tableaux E. (coissant > Tri à bulles: 11 parlonn + 3: 10: 2:1:3:1:4		Airtabu uniforme - méthoda préfere				
(th) on Endic le noe opé pour chaque (th) bille et on en rait la moyenne [t3] (comportement quand not 2 n-) (tn) ex: algo tri ale tableaux de noe entiers codre Entrée: un tableau d'entiers E, un oidre 0 sur les entiers Proposétée: le tableau d'entiers S Proposétée: le tableau d'entiers S Proposétée: le tableau d'entiers S le tableau soit trié selon l'oidre 0 le tableau soit trié selon l'oidre 0 le tableau s pent être abtenu en effectuant des échanges de valeurs entre les cases du tableau E Mesure: on effectue des analyses de complexités en fonction du nobe de cases du tableau E. (coissant / Tri à bulles: 11 parlonn + 3: 10: 2:1:3:14		ex: tableanx de taille 10				
(E2) beille et on en fait la mayenne [13] (comportement quand not 2 n-) i (tn) ex: 9190 tri ole tableaux de nove entiers codre Entrée: un tableau d'entiers E un oidre 0 sur les entiers oil, tormel sochie: un tableau d'entiers S. Propriétée: le tableau soit trié selon l'oidre 0 le tableau s pent être obtenu en effectuant des échanges de valeurs entre les cases du tableau E. Mesure: on effectue des analyses de complexités en fonction du nove de cares du tableau E. (coissant / Tri à bulles: 11 parloin 7:3:10:2:1:3:14 3:7:2:1:3:1:4:10 ok 1 2 parcours 3.7:10:2:1:3:14 3:7:2:1:3:1:4:7:10 exnange		analyse shi rows jes				
(t2) bille et on en fait la mayenne [t3] (compostement quand nex 2 n-) care plago tri de tableaux de noe entiers ex: algo tri de tableaux de noe entiers codre Entrée: un tableau d'entiers E, un oidre 0 sur les entiers Propriétée: le tableau J soit trié selon l'oidre 0 le tableaux J pent être obtenu en effectuant des échanges de valeurs entre les cases du tableaux E. Mesure: on effectue des analyses de complexités en fonction du note de cares du tableaux E. (coissant? Tri à bulles: 1'' parloun 7:3:10:2:1:3:1:4 3:1:4:10 ok 1 1 a parcours 3.7:10:2:1:3:1:4 3:7:2:1:3:1:4:7:10 échange		[t1] on étudie le noe épé pour chaque				
ex: algo tri de tableaux de nhe entiers ordre Entrée: un tableau d'entiers E, un ordre 0 sur les entiers Propriétée: le tableau 1 soit trié selon l'ordre 0 le tableau 1 pent être obtenu en effectuant des échanges de valeurs entre les cases du tableau E. Mesure: on effectue des analyses de complexités en fonction du nhe de cares du tableau E. (coissant 1 Tri à bulles: 1' parcoun 7:3:10:2:1:3:1:4 3:1:4:10 ox 3:7:10:2:1:3:1:4 3:7:2:1:3:1:4:10 ox 3:7:10:2:1:3:1:4 3:7:2:10 échange		(t2) bille et on en fait la moyenne				
ex: 919.0 tri de tableaux de nhe entiers ordre Frirée: un tableau d'entiers E, un ordre 0 sur les entiers Dochie: un tableau d'entiers S. Propriétée: le tableau s'esit trié selon l'ordre 0 le tableau s pent être obtenu en effectuant des echanges de valeurs entre les cases du tableau E. Mesure: on effectue des analyses de complexités en fonction du nhe de cases du tableau E. (coissant 7 Tri à bulles: 1'' parconn 7:3:10:2:1:3:1:4 3:1:4:10 OK 3:7:10:2:1:3:1:4 3:7:2:1:3:1:4:10 OK 3:7:10:2:1:3:1:4 3:7:2:1:3:1:4:10 OK 3:7:10:2:1:3:1:4 3:2:1:3:1:4:7:10		[£3] (comportement quand n++ & n)				
ex: algo tri de tableans de nhe entiers ordre Fintrée: un tablean d'entiers E, un oidre 0 sur les entiers Sortie: un tablean d'entiers S. Propriétée: le tableau s soit trié selon l'oidre 0 le tableau s peut être obtenu en effectuant des échanges de valeurs entre les cases du tableau E. Mesure: on effectue des analyses de complexités en fonction du nhe de cases du tableau E. (coissant / Tri à bulles: 1'' parlour 7:3:10:2:1:3:1:4						
ordre Fintrée: un tableau d'entiers E, un ordre 0 sur les entiers Sochie: un tableau d'entiers S. Proposètée: le tableau 1 soit tré selon l'ordre 0 le tableau S pent être obtenu en effectuant des échanges de valeurs entre les cases du tableau E. Mesure: on effectue des analyses de complexités en fonction du nbe de cases du tableau E. (roissant / Tri à bulles: 11 parcoun 7:3:10:2:1:3:1:4 3:1:4:10 OK 3.7:10:2:1:3:1:4 3:7:2:1:3:1:4:7:10 echange		(tn)				
ordre Fintrée: un tableau d'entiers E, un ordre 0 sur les entiers Sochie: un tableau d'entiers S. Proposètée: le tableau 1 soit tré selon l'ordre 0 le tableau S pent être obtenu en effectuant des échanges de valeurs entre les cases du tableau E. Mesure: on effectue des analyses de complexités en fonction du nbe de cases du tableau E. (roissant / Tri à bulles: 11 parcoun 7:3:10:2:1:3:1:4 3:1:4:10 OK 3.7:10:2:1:3:1:4 3:7:2:1:3:1:4:7:10 echange						
ordie Entrée : un tableau d'entiers E, un ordie 0 sur les entiers Obi, formel Sochie : un tableau d'entiers S. Propriétée : le tableau I soit tré selon l'ordie 0 le tableau S peut être obtenu en effectuant des échanges de valeurs entre les cases du tableau E. Mesure : on effectue des analyses de complexités en fonction du nbe de cases du tableau E. (roissant / Tri à bulles: 11 parcoun 7:3:10:2:1:3:1:4 3:1:4:10 OK 3:7:10:2:1:3:1:4 3:7:2:1:3:1:4:10 OK 3:7:10:2:1:3:1:4 3:2:1:3:1:4:7:10		ex. also to de tableaux de nhe inters				
Ob, former Sochie: un tablean d'entiers S. Propriétée: le tableau I soit trè selon l'oidre O le tableau I pent être obtenu en effectuant des échanges de valeurs entre les cases du tableau E. Mesure: on effectue des analyses de complexités en fonction du nbe de cases du tableau E. (coissant / Tri à bulles: 1'' parlour 7:3:10:2:1:3:1:4 3:7:2:1:3:1:4:10 ok 1 2nd parcours 3:7:10:2:1:3:1:4 3:7:2:1:3:1:4:10 ok 2 3:7:2:1:3:1:4:7:10 èchange		CA. GIND III VIC INDICATE NE NO CITATION				
Ob; Former Sochie: un tablean d'entiers S. Propriétée: le tableau soit trè selon l'ordre O le tableau s pent être obtenu en effectuant des échanges de valeurs entre les cases du tableau E. Mesure: on effectue des analyses de complexités en fonction du nbe de cases du tableau E. (coissant / Tri à bulles: 1'' parlour 7:3:10:2:1:3:1:4 3:7:2:1:3:1:4:10 ok 3:7:10:2:1:3:1:4 3:7:2:1:3:1:4:10 ok 3:7:10:2:1:3:1:4 3:2:1:4:7:10 echange	a.d.a	Furrage un tableau d'entiers E un aidre 0 sur les entiers				
Propriétée: le tableau soit trè selon l'ordre 0 le tableau s pent être obtenu en effectuant des échanges de valeurs entre les cases du tableau E. Mesure: on effectue des analyses de complexités en fonction du nbe de cares du tableau E. (coissant 7 Tri à bulles: 11 parlour 7:3:10:2:1:3:1:4 3:7:2:1:3:1:4:10 ok 3:7:10:2:1:3:1:4 3:7:2:1:3:1:4:10 ok 3:7:10:2:1:3:1:4 3:2:1:3:1:4:7:10	4.					
echanges de valeurs entre les cases du tableau E. Mesure: on effectue des analyses de complexités en fonction an nbe de cases an tableau E. (coissant / Tri à bulles: 1'' parcour) 7:3:10:2:1:3:1:4 3:7:10:2:1:3:1:4 OK 3:7:10:2:1:3:1:4 3:7:10:2:1:3:1:4 3:7:10:2:1:3:1:4 3:7:10:2:1:3:1:4 3:7:10:2:1:3:1:4 3:7:10:2:1:3:1:4 6 change	ob, formel					
echanges de valeurs entre les cases du tablean E. Mesure: on effectue des analyses de complexités en fonction du nbe de cases du tableau E. croissant 7 Tri à bulles: 1'' parcouri 7:3:10:2:1:3:1:4 3:7:2:1:3:1:4:10 ok 3.7:10:2:1:3:1:4 3:7:2:1:3:1:4:10 ok 3.7:10:2:1:3:1:4 3:2:1:3:1:4:7:10						
Mesure: on effectue des analyses de complexités en fonction du nbe de cases du tableau E. (coiscant / Tri à bulles: 1'' parcour 7:3:10:2:1:3:1:4 3 s'il y a en un échange, algo à sinon au êt 3:7:10:2:1:3:1:4 3:7:2:1:3:1:4:10 ox J. Marcours 3:7:10:2:1:3:1:4 3 d.1.3:1:4:7:10 échange		- le tablean s pent erre objent en ellectriaght des				
du nbe de cases du tableau E.		échanges de valeurs entre les cases qui tableau				
du nbe de cases du tableau E.		E.				
(roissant 7 Tri à bulles: 1" parcour) 7:3:10:2:1:3:1:4 3:7:2:1:3:1:4:10 ok 2.7:10:2:1:3:1:4 3.7:10:2:1:3:1:4 3.7:10:2:1:3:1:4 6 change 6 change 3.7:10:2:1:3:1:4 6 change						
1 ^{er} parcour) 7:3:10:2.1:3.1.4 3:7:2.1:3.1.4.10 ox 3.7.10.2.1.3.1.4 3.7.2.1.3.1.4.10 esthange 3.7.10.2.1.3.1.4 32.1.3.1.4.7:10		du noe de cases du tableau E.				
1 ^{er} parcour) 7:3:10:2.1:3.1.4 3:7:2.1:3.1.4.10 ox 3.7.10.2.1.3.1.4 3.7:2.1.3.1.4.10 echange 3.7.10.2.1.3.1.4 3.2.1.3.1.4.7:10						
échange 3.7.10.2.1.3.1.4 3.7.2.1.3.1.4.10 ox 3.7.10.2.1.3.1.4 3.2.1.3,1,4.7;10 échange	croiscant 1	Tri à bulles:				
échange 3.7.10.2.1.3.1.4 3.7.2.1.3.1.4.10 ok 3.7.10.2.1.3.1.4 3.2.1.3,1,4.7;10 échange						
échange 3.7.10.2.1.3.1.4 3.7.2.1.3.1.4.10 ok 3.7.10.2.1.3.1.4 3.2.1.3.1.4.7:10 échange	1er parlour)	7.3:10:2.1.3.1.4 3 s'il ya en un echange, algo 3				
3.7.10.2.1.3.1.4 3.7.10.2.1.3.1.4 3.7.10.2.1.3.1.4 3.7.10.2.1.3.1.4 3.7.10.2.1.3.1.4 6 change						
3.7.10.2.1.3.1.4 3.7.10.2.1.3.1.4 3.7.10.2.1.3.1.4 3.7.10.2.1.3.1.4 3.7.10.2.1.3.1.4 3.7.10.2.1.3.1.4		▲ 보고 있는데, 보고 있다. 그렇게 보고 있는데, 그런데 그런데 그렇게 되었다. 그런데, 네티워 보고 있는데, 네티워 바로 보고 있는데, 네티워 보고 있다. 그런데, 네티워 네티워 네티워 없는데,				
3.7.10.1.1.3.1.4 3 2,1.3,1,4.7;10						
3.7.10.1.1.3.1.4 3 d.1.3,1,4.7;10 echange		1 1 200 0000000000000000000000000000000				
echange						
3. + . 2 . 10 . 1 . 3 . 1 . 7						
		3.4.2.10.1.5.1.7				
		•				
		•				

V	old teiBulle (int langueur, int * T // int T[]){
	int is
	int ech fait = 1 temp;
	while (ech_tait) { ech-tait=0;
	for (i=0; i < longheur; i+t) {
	if (T[i] > T[i+1]) {
	temp=T(i);
	T(i): T[i+1];
	T [itl] = temp;
	ech_fait = 1;
	,
3	
	· les senses modifica o sur T sort des échanges de vales 4 donc la propriété (2) est vérifiée · la sondi o sur "ech-foit" nous garantie que le fable au
	trié lorsque l'algo se termine.
	propriété 1 est vérifiée
	· à chaque tour de la boncle principale, la taille du
	tablean à trier diminue de l'ase au moins.
	rataille au tableau étant finie, le tableau finit
	par être Hié.
\i\	le ube de tour boncle = longueur du tablean

	Suites numériques	et domination
Formellement	, une suite numérique es	st une application de A
R .		
exemple: 1,2	. 4. 8. 16, 32	
	n directe: Un= 2"	(préférée)
> definition	récursive: Mo= 1, M	n+2 = 2. Wh
Exemple de	nites	
· Juite const	ante: V, = 4 N: 4	4, 4, 4
· suite qua	Iranque: H: n2 = n	×n N:0,1,4,9,16,
· snite cubi	que. Non 3 - nxnx1	n U:0,1,8,27,64
· snite expor	rentielle: Un : 2h = das	2 × 2 × 2 N: 12
		K: Nan (log_(0) n'exist
V	2 2,32 2,58 2,8	
· snite racin	e carrée: La = Vn	
Rappels:		
- a b a ' = a b		
- (ab) ' - ab	46) (a1) 12 - a 2x 12 - a 2 - a
· Va . Va -	2/2 - (ra)2 = q = Va2	
- Va = 9 2/b.		
	2 = log (ba)	
On pent co	mbiner les suites de bas	es pour faire des suites
Vn = n2	K_{n-n} $\begin{pmatrix} V \\ K \end{pmatrix}_{n} = \frac{V_{n-n}}{K_{n-n}}$	l . n×h - n
A 5n3 L A		
n ·	3 n+1 p 1 n x log	10 (n) +5



	Pour denx suites quelquonque, une me domine par torien
	l'antre.
	Notations
	- S; Dest une suite on note O(D) l'ensemble des suites domine
omágn Q	- Si U est une suite on note Q(U) l'ensemble des suites qui
	- U= O(D) plutôt que U+ O(D)
	- U ((D on U= O(P) on D= Q(U) (on D>) U).
	Propriétés
	Soient U, V, W des suites (quelconques) et m, w des nbes rééls (quellonques) abous:
	- Si UKL V et VKC W Glors UKK W (transitivité)
	- Si U, V << W glors m U + V << W.
	- Si pour tout ne N (Vn & Iml I Wn alors V LL W
	- l'ensemble O(1) est l'ensemble des suites bornées
	dominées par cette suite constante (1,1,1,1)
	- L'ensemble O(V) est égale à l'ensemble (VI. O(1)
	(= {(Vn. Un) now /U suite bornéé})
	Remarques: Si UIF, W sont des suites alors
	- V LLU (réflexivité)
	- O(U)O(F) - O(F,U)
	- Si U K W alors U. Y K W. Y
	Notation: Si V e V des suites e que UC(V e V C(U) on note
: ktq	U- D(V) Lon V: D(U) c'est une relation d'équivalence)

	ecemple:
	$n < (3n + 3n < (n) \rightarrow 3n = \Theta(n)$
	Exemple: (avec n pour variable e a,b, c nber positifs non nuls avec
	-4n5+3n4+10n+20-0(n5)
	-1-0(n) mais n + 0(1)
as de	- log (n) (c n b << c n << n (e par d'égatire de classe)
ciproque du	- si a ch alors na (c nb (e par d'égalité de classe)
NSK M	no n (a-b) -> si negatif 2 (tend vers 0) donc borner
77701	- si a l b alors an l b " (2 pas d'égalité de classe)
	- log(n): 0 (fn(n))
	- nxe(n) + O(fn(n)) mais fn(n) = O(n ln(n))
)	exemple:
	Wo = 1
	Un= Un + 1 (mite léciproque)
	=> trouver forme directe pour comparer des suites définies pour
	réchrence
	M64 - M20 - M32 MAN 105 (n)
	- D (1092(n))
	W +1 +1 +1
	8 :

Algorithme non déterministe
· Un algo NP ne vérifie par forcement la règle "denx exémo
and mm entrées => algo se comporte de la nim manière
· NPA (entre 1, entré 2)
lors du debugga
DA (entrée », entree 2. suites Valeur Aléatoires) par ex
remarque: me porrespond pay à la compleme
Complexité moyenne et dans le pice des cas
(M en tops (l'espace) > pire des cas en tops (l'espace)
· (Qnick sort n new : Nhe d'apé may ou Quicksort selon n la ta
An tablean
· (Q'n): " dans le pice des cas du Enickeart
· (F'n) : " dans le pire des cas du Tri fusion
-> On a:
n·log(n)& O, F' << n·log(n) << n² << 0 ' << n²
Pourtant, "en pratique" et est + esticare que le TF.
-> TE après tests est etticace (nuelles alloca à opina suppléments
tandis que QS ne tait que des échanges de valeurs).
Pice des cas au Cs = tacheau ordonné de un sens, un vent
l'ordonner dans l'ordre inverse, nº complexité. (emment l'annéliuser)
- utiliser une valeur aléatoire comme valeur pivot Paus la
plupart des las, ga se passe bien.
Parallélisme
Un algo est dit palletisable si an mains deux soul-partiel liene-d
de l'algorithme sont indépendantes en termes d'entrée-sortie.

	Francisco de
	Exemple:
-	
-	algopaia (){
	rishitat 1 : Jons-algor-);
	résutait à : sous-algor(.); il entirées de cet appel na
	a dépendent pas de résultat à
	Ici, les appels sons-algo 2 e son-algo 2 peravent être échanges "dans
	le temps on calculé/éceuté en nême tos si en implante l'alga
	en machine

	Listes chaînées					
	Mne liste chaînée est un ensemble de zoner dispusées dans la mêmoire, toutes ayant la mêmo taille et chainne contient: - 1 valeur - 1 adroix vers la cellule "suivante" (logiquement)					
	Remarque: paris un tableau, tontes les zouver sont contigués.					
	Exemple					
	En C:					
ty pe det	struct s(ellule t { int val; // valeur struct s(ellule t * suiv; // valeur suivante } cellule _t;					
	lemarque. Une liste vide sera représentée par la valeur NULL.					
include stability	Construction d'une liste taille 1 (cellule to cons(int v, cellule to queue) ((cellule to recellule to malloc (size of (*r)); if (s== NVLL) exit (1);					
(typeast val)	(*r) val= v; //r-> val= v; (*r) . suiv = queve; // r-> val= v; (eturn r;					

Appel de la 1	conchion:			
cellule +* 1	ifte : NULL:			
liste - cons	(3, lite).			
liste = cons	(2, liste);			
fonction d'a	fichage:			
void afficher	(cellule _t * 1)			
	FNVLL) {			
0(intf ("% d" (* 1)	.val);		
	1-> suiv;	•		
}	,			
printf ("	m").			
fonction libéra	tion espace (por	r malloc()):		
void libérer	cellule + = e)	{		
	VULL) return;			
liberer (1 > saivi			
F128(P).	l- NULL;			
}				
Exemple:				
E	200-1011	da. (1)		
	2 appel Gali	1-175 fe)		
	1	5 2nd uppel		
	1/4	3 3 a	PP-1	
	1/1	(while		
	1+	+ X		
				2.50