

TD 2 :

TD Pointeur

Ex 1:

ecrire(*pi) \Rightarrow 5

ecrire(*pj) \Rightarrow 6

ecrire(*pi) \Rightarrow 5

ecrire(*pj) \Rightarrow 5

pi A10 @A10

i 5

pj A20

j 6 @A20

Ex 2:

	a	b	c	P1	P2
valeur initiale	10	15	20	NULL	NULL
P1 \leftarrow &a				@a	@c
P2 \leftarrow &c					
*P1 \leftarrow *P2 + 1	21	15	20		
P1 \leftarrow P2				@c	@c
P2 \leftarrow &b				@c	@b
*P1 \leftarrow *P1 * *P2			5		
*P2 \leftarrow *P2 + 1		16			
*P1 \leftarrow *P1 ** *P2			10		
P1 \leftarrow &a				@a	

Ex 3:

P	P+1	P+2
2	5	8
9	12	13
15	18	22
25		

$$*P + 2 \Rightarrow 2 + 2 = 4$$

$$*(P+2) \Rightarrow (*tab[3]) \Rightarrow tab[3] = 8$$

&P+1 Pointeur sur pointeur sur entier

$$*tab[4] - 3 \Rightarrow *tab[7] = tab$$

$$tab + 3 \Rightarrow \&tab[4]$$

$$\&tab[7] - P \Rightarrow tab + 6 - tab \Rightarrow 6$$

$$P + (*P - 10) \Rightarrow tab + (2 - 10) \Rightarrow tab - 8$$

\Rightarrow pointeur

$$*(P + *(P + 8) - tab[7])$$

$$\Rightarrow *(P + 22 - 15) \Rightarrow *(tab + 7) \Rightarrow 18$$

Ex 4:

Procédure remplir(T: tab, N: entier)

variable

P: entier

Debut

Pour P de T à T+N-1 faire

ecrire("donner T", P-T+1, "J")

lire(*P)

Fin pour

Fin

Procédure affiche(T: tab, N: entier)

variable

P: entier

Debut

Pour p de T à T+N-1 faire

ecrire(*p)

Fin pour

Fin

Procédure Tri(T: tab, N: entier)

variable

P: entier

Permut: booléen

aux: entier

Debut

Repete

Permut \leftarrow faux

Pour p de T à T+N-1 faire

si (*p > *(p+1) alors

```

aux ← *p
*p ← *(p+1)
*(p+1) ← aux
Permut ← vrai

```

Finsi

Jusqu'à (Permut = faux)

Fin

Algorithme ex4

Type

tab : tableau [1..100] d'entiers

variable

T: tab

N: entier

debut

ecrire ("donner la taille")

lire (N)

remplir (T, N)

tri (T, N)

Affiche (T, N)

Fin

Ex 5:

ouvrier = Enregistrement

Nom : chaîne [20]

Anc : entier

sal : réel

Fin enreg

TO : tableau [1...200] d'ouvriers

Procédure saisie (T:TO, N: entier)

variable

P: *ouvrier

Debut

Pour P de T à T+N-1 faire

Repete

lire ((*P).nom)

Jusqu'à ((long-(*P).nom) < 20)

lire (P → Anc)

lire (*P.sal)

Finsour

Fin

Procédure Tri (T:TO, N: entier, R:TO)

variable

debut

P ← R, q ← R+N-1

Repete

Pmax ← cherche-Max (T, N)

*P ← *Pmax

supprimer (T, N, Pmax)

P ← P+1

Pmin ← cherche-min (T, N)

*q ← *Pmin

supprimer (T, N, Pmin)

q ← q-1

Jusqu'à (P=q)

Fonction cherche-Max (T:TO, N:entier)

: *entier

var

P: *entier

debut

Pmax ← T

Pour P de T+1 à T+N-1 faire

si (*P).sal > (*Pmax).sal alors

Pmax ← P

Finsi

Finsour

retourner (Pmax)

Fin

Ex 7: (TD Enregistrement)

Adresse = Enregistrement

code: Entier

ville: chaîne de 8 caractères

Pays: chaîne [15]

fin enreg

client: Enregistrement

Nom: chaîne [20]

Prenom: chaîne [20]

Adr: adresse

Mt: reel

fin Enreg

Entreprise: tableau [1..100] de client

Procédure remplissage (var c: client)

var

debut

lire (c.Na)

lire (c.prenom)

lire (c.Adr.code)

lire (c.Adr.ville)

lire (c.Adr.Pays)

lire (c.Mt)

fin

Procédure remplir (T: Entreprise, N: entier)

variable

P: *client

Debut

Pour P de T à T+N-1 faire

Remplissage (*P)

finpour

fin

Procédure affiche (c: client)

var

debut

écrire ("Nom:", c.Nom)

écrire ("Prenom:", c.Prenom,

"Adresse", c.Adr.code, c.Adr.ville,

c.Adr.Pays,

"Montant:", c.Mt)

fin

Procédure Affichage (T: *client, N: entier)

variable

debut

Pour P de T à T+N-1 faire

Affiche (*P)

finpour

fin

fonction aléat (E: Entreprise, N: entier)

: client

var

P: Pmax

Debut

Pmax ← E

Pour P de E+1 à E+N-1 faire

si (P → Mt) > (Pmax → Mt) alors

Pmax ← P

fin si

finpour

retourner (*Pmax)

fin

fonction Tunisie (c: client)

variable

ch1, ch2: chaîne [15]

P, q: *caractère

Debut

ch1 ← c.Adr.Pays

ch2 ← "Tunisie"

si (long-ch(ch1) < long-ch(ch2)) alors

retourner (faux)