

الفصل 5 : الجداول وسلاسل الرموز

1. مقدمة

في البرمجة، يتم تنظيم البيانات على شكل ثوابت ومتغيرات بطرق معينة، لتسهيل معالجتها، والوصول إليها بسرعة أكبر. وللبيانات أنواع مختلفة، يمكن تقسيمها إلى قسمين : الأنواع البسيطة. مثل: الأعداد الصحيحة، الأعداد الحقيقية، الرموز، والنوع المنطقي. والأنواع المركبة: وهي الجداول والبنى أو السجلات.

لنفرض أننا نريد ادخال علامات 1000 طالب، لتحليلها وحساب بعض الإحصائيات. في هذه الحالة، سيكون من غير المعقول استعمال 1000 متغير لحفظ العلامات، ومن ثم كتابة 1000 عملية ادخال في البرنامج. من الأفضل استعمال متغير واحد، يمكنه أن يحوي جميع قيم العلامات، واستعمال حلقة لإدخالها. وتسمى البنية التي يمكنها تخزين عدة قيم في نفس الوقت بالجدول.

في هذا الدرس، سوف نتطرق إلى الجداول الثابتة (statique) بنوعها، أحادية البعد، ومتعددة الأبعاد. كما سنرى السلاسل التي تُعتبر حالة خاصة من الجداول.

2. النوع جدول

2.1. التعريف

الجدول (بالفرنسية tableau بالإنجليزية array): هو بنية بيانات مركبة، تتكون من مجموعة محدودة من العناصر المتجانسة (من نفس النوع)، والتي يمكن الوصول إليها بواسطة مؤشرات تشير إلى موقعها. يمكن رؤية الجدول على أنه مجموعة من المتغيرات، من نفس النوع، لها نفس الاسم. البعد: بُعد جدول هو عدد المؤشرات الضرورية لتحديد عنصر واحد.

المؤشر: عندما يتم تخزين البيانات في جدول، يتم تحديد العنصر بواسطة مؤشر، وهو، في لغة C، عبارة عن عدد صحيح غير سالب ($0 \leq$). يبدأ المؤشر من 0 حتى $N - 1$ (حيث N هو حجم الجدول).

جدول أحادي البعد

ويسمى أيضا بالشعاع (vecteur): يمكن الوصول إلى أي عنصر من عناصره بواسطة مؤشر واحد، حيث كل قيمة للمؤشر تُحدد عنصرا من الجدول.

2.2. التمثيل

ويتم تمثيل الجدول في الذاكرة على شكل تسلسل من الخلايا المتجاورة. ولا يمكن حذف او اضافة خلية جديدة إلى الجدول بعد إنشائه (statique). يمثل الشكل التالي جدولا من 8 أعداد حقيقية.

7	6	5	4	3	2	1	0	المؤشر
3-	0	2	9	0	3-	7	15	القيمة

لا يوجد فرق بين رسم الجدول عموديا أو أفقيا.

2.3. التصريح

لغة C	الخوارزم
typeElements nomTableau [Taille] ;	nomTableau [Taille] : Tableau de typeElements نوع العناصر [عدد العناصر] : Tableau de

- حيث ان كلمة Tableau de هي كلمة محجوزة في الخوارزم، تُستعمل لتدلّ أنّ المتغيّر عبارة عن جدول.
- nomTableau اسم الجدول: هو اسم المُعرّف الذي يطلق على الجدول (اسم المتغيّر).
 - Taille : عدد العناصر
 - typeElements : نوع العناصر، يمكن أن يكون أيّ نوع مثل entier(int)، réel(float)، ...
- للتصريح بعدة جداول من نفس النوع نستعمل الفاصلة "،" مع تحديد حجم كل جدول بين عارضتين.

مثال

const int N=100 ; float note[N] ; int tab1[50],tab2[20];	const N=100 note[N] : tableau de réel tab1[50],tab2[20] : tableau d'entier
--	--

2.4. التهيئة initialization

في C، يمكن تحديد قيم ابتدائية لجميع عناصر الجدول داخل حاضنتين { و } أثناء التصريح بالجدول. حيث يتم فصل القيم باستخدام الفاصلة "،"، ويجب أن تكون هاته القيم من نفس النوع.

مثال

```
int tab[] = { 15, 7, -2, 0, 9, 2, 0, -3};
```

7	6	5	4	3	2	1	0	المؤشر
3-	0	2	9	0	2-	7	15	القيمة

ملاحظة: يمكن تحديد عدد العناصر بين العارضتين، أو تركها فارغة ليتمّ حسابها أوتوماتيكياً.

2.5. الاستعمال

لا يمكن التعامل مع الجدول ككتلة واحدة مثل $tab \leftarrow 10$ ، وإنما يجب التعامل مع كل عنصر على حدة. وللوصول إلى عنصر واحد من الجدول، نستخدم اسم الجدول مع مؤشر داخل عارضتين [و]، وعند حساب العبارة التي بين العارضتين نتحصل على قيمة من نوع عدد صحيح. للوصول إلى العنصر الأول من الجدول tab ، نستخدم $tab[0]$. وللوصول إلى العنصر الرابع، نستخدم $tab[3]$.

مثال $tab[5-3] \leftarrow tab[tab[3]+1] \Leftrightarrow tab[2] \leftarrow tab[0+1] \Leftrightarrow tab[2] \leftarrow 7$

ليصبح الجدول

7	6	5	4	3	2	1	0	المؤشر
3-	0	2	9	0	7	7	15	القيمة

ملاحظة: محاولة الوصول لعنصر غير موجود (في حالة المؤشر أكبر أو يساوي حجم الجدول أو سالب) يؤدي إلى تعطل البرنامج.

2.6. قراءة جدول

لملء جدول يتكوّن من N عدد، نستعمل "lire" N مرّة مثل:

لغة C	الخوارزم
scanf("%d", &tab[0]);	lire(tab[0])
scanf("%d", &tab[1]);	lire(tab[1])
...	...
scanf("%d", &tab[N-1]);	lire(tab[N-1])

ولكن، نلاحظ أنّ التعليمة lire تتكرّر N مرّة. أي، تتكرّر من 0 إلى N-1. لذلك يمكن استعمال الحلقة pour. حيث يلعب العدّاد دور المؤشر.

لغة C	الخوارزم
<pre>for(i = 0; i < N; i++){ printf("nb %d =>", i); // للتوضيح فقط scanf("%d", &tab[i]); }</pre>	<pre>pour i←0 à N-1 faire écrire("nb", i, "=>")// للتوضيح فقط lire(tab[i]) finPour</pre>

التعليمة lire لملء الجدول، أما écrire فهي لتوضيح للمستعمل ما المطلوب منه.

2.7. طباعة جدول

مثل القراءة، écrire تتكرر.

لغة C	الخوارزم
<pre>for(i = 0; i < N; i++){ printf("%d\t", tab[i]); }</pre>	<pre>pour i←0 à N-1 faire écrire(tab[i]) finPour</pre>

2.8. ملاحظات

لزيرة جميع عناصر جدول، نستعمل الحلقة pour.

يجب تحديد حجم الجدول اثناء البرمجة (التصريح)، ولكن، يمكن ان نوهم المستعمل أن حجم الجدول يتغير، وذلك بالتصريح بجدول كبير، ثم نستعمل جزءا منه فقط. ونطلب من المستعمل الحجم الذي يريده، شريطة ألا يتعدى الحجم الحقيقي للجدول.

2.9. مثال

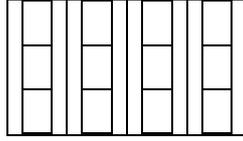
اكتب البرنامج الذي يُدخل معدّلات N طالب، حيث N يحددها المستعمل، ثم يحسب عدد الطلبة الذين لم يتحصّلوا على المادة. (معدّل اقل من 10).

لغة C	الخوارزم
<pre>#include<stdio.h> #define MAX 200 int main() float note[MAX] ; int i, N, aj=0; // aj càd nb ajournés // récupérer le nbr des étudiants do{ printf("entrer le nbr des étudiants (<%d)",MAX) ; scanf("%d",N) ; }while (N>MAX) ; // Remplir le tableau for(i = 0; i < N; i++){ printf("note %d =>", i); scanf("%d", &note[i]); } // calculer le nbr des ajournés for(i = 0; i < N; i++) if(note[i]<10) aj++ ; // affichage du resultat. printf("le nbr des ajournés est %d", aj); }</pre>	<pre>Algorithme nb_ajourne Const MAX=200 var note[MAX] :tableau de réel i, aj, N :entier début Faire écrire("entrer le nbr des étudiants (<", MAX, ")") lire(N) TQ N>MAX pour i←0 à N-1 faire écrire("note ", i, "=>") lire(note[i]) finPour aj ←0 pour i←0 à N-1 faire si note[i]<10 alors aj←aj+1 finSi finPour écrire("le nbr des ajournés est ", aj) fin</pre>

3. الجداول متعدّدة الأبعاد

3.1. التعريف

جدول ثنائي الأبعاد (يسمى أيضاً المصفوفة): هو في الواقع جدول عاديّ (أحاديّ البعد)، تكون عناصره، في حد ذاتها، عبارة عن جداول أحاديّة البعد. نرى هذا في الرسم التوضيحي بالأسفل،



يتم الوصول إلى العناصر من خلال مؤشريْن، الأول يحدّد رقم السطر، والثاني يحدّد رقم العنصر في هذا السطر (العمود). يمكن تعميم هذه الآلية لإنشاء مصفوفات لها أكثر من بُعدين. يمكننا إنشاء مصفوفة ذات n بُعد، ويتطلب الوصول إلى عناصرها n مؤشراً.

ترتيب المؤشرات أمر بالغ الأهمية. فالعنصر $M[3][2]$ (36) يختلف عن العنصر $M[2][3]$ (28).

3.2. التمثيل

ويتمّ تمثيل المصفوفة، في الذاكرة، على شكل تسلسل من الخلايا المتجاورة. ولا يمكن حذف أو إضافة خلية جديدة إلى المصفوفة بعد انشاءها. يمثل الشكل التالي مصفوفة ذات 3 أسطر و 5 أعمدة من الأعداد الحقيقية.

		رقم العمود					
		4	3	2	1	0	
رقم السطر	0	9	0	3-	7	15	0
	1	85	33	4	12	6	1
	2	52-	28	17	8-	2	2
	3	12-	49	36	42	14	3

3.3. التصريح

الخوارزم	لغة C
Tableau de typeElements : nomMatrice [Lignes][Colonnes] ; نوع العناصر : Tableau de [عدد اعمدة] [عدد اسطر] اسم المصفوفة	typeElements nomMatrice [Lignes][Colonnes] ;

حيث أنّ كلمة Tableau de هي كلمة محجوزة في الخوارزم، تستعمل لتدلّ أنّ المتغيّر عبارة عن جدول.

– nomMatrice : اسم المصفوفة هو اسم المُعرّف الذي يُطلق على المصفوفة (اسم المتغيّر).

– Lignes : عدد الأسطر

– Colonnes : عدد الأعمدة

– typeElements : نوع العناصر. يمكن أن يكون أيّ نوع. مثل: entier(int)، réel(float)، ...

عدد العناصر هو عدد الأسطر × عدد الأعمدة.

مثال

<pre>const int L=100 , C=200 ; float M[L][C] ; int mat1[50][30],mat2[30][20];</pre>	<pre>const L=100 const C=100 M[L][C] : tableau de réel mat1[50][30],mat2[30][20] : tableau d'entier</pre>
---	---

3.4. التهيئة initialization

في C، يمكن تحديد قيم ابتدائية لجميع عناصر المصفوفة بتحديد عناصر كل سطر داخل حاضنتين { و }، أثناء التصريح بالمصفوفة. حيث يتم فصل القيم باستخدام الفاصلة ",", وفصل كلّ سطر باستخدام الفاصلة ".". ويجب أن تكون جميع القيم من نفس النوع.

مثال

```
int mat[][] = {{15, 7, -3, 0, 9},{6, 12, 4,33,85},{2, -8, 17 ,28,-52},{14, 42, 36, 49, -12}};
```

رقم العمود

	4	3	2	1	0	
	9	0	3-	7	15	0
	85	33	4	12	6	1
	52-	28	17	8-	2	2
	12-	49	36	42	14	3
						رقم السطر

ملاحظة يمكن تحديد عدد الأسطر وعدد الأعمدة بين العارضتين، أو تركها فارغة ليتم حسابهما أوتوماتيكيا.

3.5.3 الاستعمال

للوصول إلى عنصر واحد من المصفوفة، نستخدم اسم المصفوفة مع مؤشر داخل عارضتين [و] يحدّد رقم السطر، ومؤشر آخر داخل عارضتين [و] يحدّد رقم العمود. للوصول إلى العنصر الموجود في السطر الأول والعمود الأول من المصفوفة mat، نستخدم mat[0][0].

mat[العمود][السطر]

الصيغة

mat[1][3]←mat[1][3]+2

مثال

ليصبح الجدول

						رقم العمود
	4	3	2	1	0	
	9	0	3-	7	15	0
	85	35	4	12	6	1
	52-	28	17	8-	2	2
	12-	49	36	42	14	3
						رقم السطر

3.6.3 قراءة مصفوفة

لملء مصفوفة M[L][C]، نقوم بملء L سطر. حيث أنّ كلّ سطر عبارة عن جدول أحادي البُعد، به C عنصر.

لغة C	الخوارزم
<pre>for (j=0 ;j<C ;j++) scanf("%d", &M[0][j]); for (j=0 ;j<C ;j++) scanf("%d", &M[1][j]); ... for (j=0 ;j<C ;j++) scanf("%d", &M[L-1][j]);</pre>	<pre>pour j←0 à C-1 faire lire(M[0][j]) fpour pour j←0 à C-1 faire lire(M[1][j]) fpour ... pour j←0 à C-1 faire lire(M[L-1][j]) fpour</pre>

ولكن، نلاحظ أنّ التعليمة pour تتكرّر L مرة. أي تتكرّر من 0 إلى L-1، لذلك يمكن استعمال الحلقة pour.

لغة C	الخوارزم
<pre>for(i = 0; i < L; i++) for(j = 0; j < C; j++){ printf("M[%d, %d] =>", i,j); // للتوضيح فقط scanf("%d", &M[i][j]); }</pre>	<pre>pour i←0 à L-1 faire pour j←0 à C-1 faire // للتوضيح فقط écrire("M[" , i, ", ", j, "]=>") lire(M[i][j]) finPour finPour</pre>

التعليمة "lire" لملء الجدول، أما "écrire" فهي لتوضيح للمستعمل ما المطلوب منه. تحتوي التعليمة for(j... الداخلية على تعليمتين printf للشرح، و scanf لإدخال القيم. أما for(i... الخارجية فتحتوي على تعليمة واحدة هي for(j... .
شرح écrire التوضيحية: نفرض ان i=3 و j=5

écrire("M["	i	,"	j	"]=>"
الشاشة	M[3	,	5]=>

3.7. طباعة مصفوفة

مثل القراءة، écrire تتكرر.

لغة C	الخوارزم
<pre>for(i = 0; i < L; i++){ for(j = 0; j < C; j++) printf("%d\t", M[i][j]); printf("\n"); }</pre>	<pre>pour i←0 à L-1 faire pour j←0 à C-1 faire écrire(M[i][j]) finPour finPour</pre>

تحتوي التعليمة for(j... الداخلية على تعليمة واحدة printf لطباعة العنصر M[i][j]. أما for(i... الخارجية فتحتوي على تعليمتين for(j... لطباعة السطر i، و printf("\n") للرجوع إلى السطر في نهاية كل سطر i من المصفوفة.
ملاحظة: لزيارة جميع عناصر المصفوفة نستعمل حلقتين pour.

3.8. مثال

اكتب البرنامج الذي يدخل درجات الحرارة لكل ساعة، لمدة 30 يوماً، على شكل مصفوفة (30 على 24)، ثم يظهرها على الشاشة، وبعد ذلك يظهر أكبر درجة حرارة، ومتى كانت.

لغة C	الخوارزم
<pre>#include<stdio.h> #define Jr 30 // nb lignes #define Hr 24 // nb colonnes int main(){ float T[Jr][Hr], maxT; // max température int i, j, maxJr, maxHr; // Remplir les températures for(i = 0; i < Jr; i++){ for(j = 0; j < Hr; j++){ printf("T[%d, %d] =>", i+1, j); // للتوضيح فقط scanf("%d", &T[i][j]); } } // afficher toutes les températures for(i = 0; i < Jr; i++){ for(j = 0; j < Hr; j++) printf("%d\t", M[i][j]); printf("\n"); } // recherche de la température maximale maxT=T[0][0]; maxJr=0; maxHr=0;</pre>	<pre>Algorithme nb_ajourne Const Jr =30 Hr=24 var T[Jr][Hr] :tableau de réel maxT :réel i, j, maxJr, maxHr :entier début pour i←0 à Jr-1 faire pour j←0 à Hr-1 faire écrire("T[" , i+1, ", ", j, "]=>")// للتوضيح فقط lire(T[i][j]) finPour finPour pour i←0 à Jr-1 faire pour j←0 à Hr-1 faire écrire(M[i][j]) finPour finPour maxT←T[0][0] maxJr←0 maxHr←0</pre>

<pre> maxHr=0 ; for(i = 0; i < Jr; i++) for(j = 0; j < Hr; j++) if (T[i][j]>maxT){ maxT=T[i][j]; maxJr=i; maxHr=j; } //affichage des résultats printf("la température maximale est %d et a été enregistrée le %d à %d", maxT, maxJr+1, maxHr) ; </pre>	<pre> pour i←0 à Jr-1 faire pour j←0 à Hr-1 faire si (T[i][j]>maxT) alors maxT←T[i][j] maxJr←i maxHr←j finSi finPour finPour écrire("la température maximale est ", maxT," et a été enregistrée le ", maxJr+1, " à ", maxHr) fin </pre>
---	---

البرنامج يأخذ مصفوفة T من درجات الحرارة، ويرجع لنا أكبر درجة حرارة مسجلة maxT، واليوم الذي سُجّلت فيه maxJr، والساعة التي سُجّلت فيها maxHr. بعد تعبئة المصفوفة واطهارها، نفرض أنّ أكبر درجة حرارة هي الموجودة في السطر 0 والساعة 0، ثمّ نقوم بزيارة جميع عناصر المصفوفة، وكلّما وجدنا درجة حرارة أكبر من المخزّنة في maxT، تتغيّر maxT. وبالتالي تتغيّر maxJr و maxHr. وفي الأخير نظهر النتائج على الشاشة، مع زيادة 1 لـ maxJr لأنّ السطر يبدأ من الصفر 0، واليوم يبدأ من الواحد 1.

4. سلاسل الرموز

4.1. التعريف

سلسلة رموز: Chaines de caractères String هي مجموعة مرتّبة من الرموز، يكون طولها 0 أو أكثر. وتكون دوماً بين علامتي اقتباس ثنائية <> مثل "informatique"، "Bonne chance\n"، "1"، "3.14". وفي لغة C يتمّ استعمال جداول من نوع char لإنشاء السلاسل. وحين قراءة سلسلة رموز من لوحة المفاتيح، يتمّ وضع كل رمز في خانة، وحين الانتهاء من الرموز، يتمّ إضافة الرمز '\0' إلى نهاية النص للدلالة على نهايته. ويُطلق على الرمز '\0' اسم null والذي يحمل الشفرة 0. ويوجد ثابت مصرح به في المكتبة stdio.h يدعى NULL بأحرف كبيرة.

```
#define NULL 0
```

ملاحظة: بما أنّ كلّ رمز له شفرة، مثلاً شفرة 'A' هي 65، شفرة 'a' هي 97، كذلك للرمز '\0' له شفرة وهي 0. إذن،

$$0 \Leftrightarrow '\0' \Leftrightarrow \text{NULL}$$

4.2. التصريح

في الخوارزم، نستعمل النوع chaine de caractères، أمّا في C، فنستعمل جدولاً من الرموز. لنفرض أنّه لدينا السلسلة str، التي يمكنها أن تحتوي على 30 رمز على الأكثر، بما فيها '\0'. يتمّ التصريح بها كما يلي:

<pre>char str[30] ;</pre>	<pre>var str :chaine de caractères var str[30] :caractères</pre>
---------------------------	--

4.3. التصريح مع التهيئة initialization

في المثال التالي، نقوم بإنشاء جدول رموز وتهيئته بالكلمة "Welcome".

```
char slt[] = {'W', 'e', 'l', 'c', 'o', 'm', 'e', '\0'};
```

حيث تُنشئ هذه التعليمة جدولاً من 8 رموز (7 خانات لكلمة "Welcome"، وخانة لاحتواء الرمز '\0'). لكن، هناك طريقة أسهل وأسرع لإنشاء وتهيئة السلسلة وهي:

```
char slt[] = "Welcome";
```

تؤدي إلى نفس النتيجة، وهي إنشاء جدول من 8 رموز، ينتهي بالرمز '\0'.

	0	1	2	3	4	5	6	7
slt	W	e	l	c	o	m	e	\0

كما يمكن تحديد حجم الجدول:

```
char slt[30] = "Welcome";
```

	0	1	2	3	4	5	6	7	8	...	28	29
slt	W	e	l	c	o	m	e	\0	...			

4.4. الإسناد

بما أن السلاسل عبارة عن جداول، فلا يمكن إسناد سلسلة لمتغير مباشرة بعد التصريح به. والعمليّة التالية خاطئة

```
char slt[30];
```

```
slt = "Welcome"; // erreur : affectation a un tableau.
```

ولإسناد سلسلة لمتغير أو نسخ متغير إلى متغير آخر، نستعمل الدالة strcpy().

```
strcpy(slt , "Welcome" );
```

4.5. اظهار سلسلة الرموز

يمكن استعمال الصيغة %c لإظهار السلسلة حرف بحرف، إلى أن نصل إلى الرمز '\0'. مثل:

```
for (i=0 ;str[i] != '\0' ;i++)
    printf("%c",str[i]) ;
```

```
i ← 0
TQ str[i] ≠ '\0' faire
    écrire(str[i])
    i ← i+1
finTQ
```

كما يمكن اظهار السلسلة مباشرة باستعمال الصيغة %s.

```
printf("%s",str) ;
```

```
écrire(str)
```

4.6. قراءة سلسلة

يمكن ادخال السلسلة مباشرة باستعمال الصيغة %s، ودون استعمال & قبل اسم الجدول.

```
scanf("%s",str) ;
```

```
lire(str)
```

```
#include<string.h>
...
gets(slt) ;
```

ولإدخال نصوص تحتوي على فراغات، في C، نستعمل التعليمة gets المعرفة في المكتبة string.h، لأن scanf تتوقف عند أول فراغ.

4.7. بعض الدوال الخاصة بالسلاسل

يدعم C مجموعة واسعة من الدوال التي تتعامل مع السلاسل، وهي معرفة في المكتبة string.h. من بينها:

strcpy(s1, s2);	ينسخ السلسلة s2 إلى سلسلة s1.
strcat(s1, s2);	لإضافة سلسلة s2 في نهاية السلسلة s1.
strlen(s1);	تُرجع هذه الدالة طول السلسلة s1.
strcmp(s1, s2);	تُرجع 0 إذا كان s1 و s2 متماثلين ؛ وعددا أقل من 0 إذا كان s1 < s2 ؛ أكبر من 0 إذا كان s1 > s2.

4.8 امثلة

مثال 1: السلسلة الفارغة ; str="" والتي طولها 0

	0	1	2	3	4	5	6	...	28	29
str	\0							...		

لا يهّم محتوى الخانات بعد '\0'، فالسلسلة تنتهي عند أول '\0'. إذن، يمكن تحويل أيّ سلسلة إلى سلسلة فارغة بوضع str[0]='\0'

مثال 2: سلسلة تحتوي حرفا واحدا، فهي تختلف عن النوع رمز. إذن، 'w' ≠ 'w' ف "w" عبارة عن جدول.

	0	1	2	3	4	5	6	...	28	29
str	w	\0						...		

مثال 3: اكتب البرنامج الذي يقوم بإدخال نص، ثم يقوم بتحويل الحروف الكبيرة إلى صغيرة، والصغيرة إلى كبيرة.

لغة C	الخوارزم
<pre>#include<stdio.h> #include<string.h> int main(){ char txt[200] ; int i ; printf("entrer un texte\n") ; gets(txt) for(i=0 ;txt[i] !='\0' ;i++) if (txt[i]>='A' && txt[i]<='Z') txt[i]+='a'-'A' ; else if (txt[i]>='a' && txt[i]<='z') txt[i]-='a'-'A' ; printf("%s",txt) ; return 0 ; }</pre>	<pre>algorithme inverse var txt :chaine[200] i :entier début écrire(("entrer un texte")) lire(txt) i←0 TQ txt[i]≠'\0' faire si txt[i]>='A' et txt[i]<='Z' alors txt[i]=txt[i]+'a'-'A' sinon si (txt[i]>='a' et txt[i]<='z') alors txt[i]=txt[i]-('a'-'A') ; finSi finSi finTQ écrire(txt) fin.</pre>