Les chaînes de caractères

I) Le type chaîne de caractères

Une chaîne de caractère est constituée de caractères. On peut la représenter dans une expression sous la forme de la suite de caractères, placée entre apostrophe (simple ou double).

```
-->'Une chaîne de caractères'  
ans = Une chaîne de caractères  
-->"Une autre"  
ans = Une autre
```

Il faut doubler les apostrophes pour les considérer dans une chaîne

```
-->'L"été'  
ans = L'été
```

II) Opérations sur les chaînes de caractères

La fonction *length()* appliquée à une chaîne fournit le nombre de caractères de celle-ci.

```
-->s='Une chaîne de caractères'  
s = Une chaîne de caractères  
-->length(s)  
ans = 24.
```

La fonction de *concaténation* de chaînes de caractères est représentée par le symbole +.

```
-->s+' plus longue'  
ans = Une chaîne de caractères plus longue
```

La fonction *string()* permet de convertir une valeur numérique en une chaîne de caractères

```
-->a=string(2+3)  
a = 5  
-->b=string(1/3)  
b = .3333333  
-->b+a  
ans = .33333335
```

La fonction *evstr()* permet de convertir une chaîne de caractères en une expression :

```
-->c='sqrt(3)/2'  
c = sqrt(3)/2  
-->d=evstr(c)  
d = .8660254
```

La fonction *part()* permet d'extraire une sous chaîne grâce à un nombre ou un tableau

```
-->part('abcdefg',2)  
ans = b  
-->part('abcdefg',[1 2 6])  
ans = abf
```

Le fonction *strindex()* permet de rechercher une chaîne à l'intérieur d'une autre chaîne. La fonction fournit les indices de début de toutes les occurrences de la chaîne recherchée.

```
-->strindex('Mississippi','ss')  
ans = ! 3. 6. !  
-->strindex('Mississippi','i')  
ans = ! 2. 5. 8. 10. !
```

III) Définition d'un caractère

1) Les caractères en Scilab

Scilab utilise une convention qui lui est propre pour représenter les caractères. Il est possible de connaître cette représentation au moyen de la fonction *str2code* :

```
-->str2code('abc')
ans =
! 10. !
! 11. !
! 12. !
```

Réciproquement, il est possible de synthétiser une chaîne de caractères à partir d'un tableau contenant les codes des caractères que l'on désire obtenir :

```
-->code2str([10 11 12])
ans = abc
```

De manière plus générale, on pourra utiliser cette représentation d'une chaîne de caractère chaque fois que l'on désirera manipuler, transformer cette chaîne. Ainsi, on remarquera que les lettres minuscules sont représentées par les entiers consécutifs de 10 à 35 et que les lettres majuscules sont représentées par l'opposé de ces mêmes codes :

```
-->str2code('azAZ')
ans =
! 10. !
! 35. !
! - 10. !
! - 35. !
```

2) La représentation ASCII (American Standart Codes for Information Interchange)

La codification la plus utilisée dans le monde est la représentation ASCII dont le tableau suivant donne le code ASCII standart, comportant 128 caractères numérotés de 0 à 127. On utilise en Europe une extension du code ASCII, l'ISO-latin qui comporte 256 caractères, les codes de 192 à 255 contenant la plupart des caractères spécifiques des langues indo-européennes (lettres accentuées).

32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
	!	"	#	\$	%	&		(	)	*	+	,	-	.	/	0	1	2	3	4	5	6	7
56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
8	9	:	;	<	=	>	?	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_	`	a	b	c	d	e	f	g
104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

Scilab permet la conversion vers l'ASCII (et réciproquement) au moyen de la fonction *ascii()* ; cette opération est utile lorsqu'il s'agit de manipuler des textes externes à Scilab (par exemple, lus dans un fichier).

```
-->ascii('abc')
ans = ! 97. 98. 99. !
-->ascii([97 98 99])
ans = abc
```

3) Manipulation de caractères

Lecture de caractères

La fonction `readc_()` permet la lecture d'une chaîne de caractères (et donc d'un seul caractère).

L'exécution du programme courant est suspendu jusqu'à ce qu'une chaîne de caractère (éventuellement vide) suivie de la touche entrée soit donnée par l'utilisateur.

```
-->c='o';  
-->while c=='o' do  
-->  
-->  
-->disp('Voulez-vous continuer ? (o/n) ')  
-->c=readc_();  
-->end
```

Exercices

Chiffres Romains

Il s'agit de créer la fonction `dec2rom()` permettant de donner l'écriture en chiffres romains d'un nombre entier.

Lettre	Valeur
<i>M</i>	1000
<i>D</i>	500
<i>C</i>	100
<i>L</i>	50
<i>X</i>	10
<i>V</i>	5
<i>I</i>	1

On n'oubliera pas que si 3 se note III, 4 se note IV, 400 CD, ...

Réciproquement, créer la transformation inverse dans la fonction `rom2dec()`.

Palindromes

Ecrire une fonction `palindrome()` reconnaissant si la chaîne de caractères qui lui a été passé comme paramètre est un palindrome, c'est-à-dire une phrase qui se lit de la même manière (à la ponctuation près), à l'endroit et à l'envers. (Vous pourrez supposer, dans un premier temps, que la chaîne a été transformée en une chaîne constituée des majuscules non accentuées des caractères de la chaîne initiale et que les espaces ainsi que la ponctuation de cette chaîne ont été supprimés).

Quelques palindromes sont bien connus :

- Esope reste ici et se repose.
- Elu par cette crapule.
- A man, a plan, a canal : Panama.
- Georges Pérec (1936 - 1982) est l'auteur d'un palindrome de 1 247 mots et plus de 76 000

caractères, qui débute ainsi :

Trace l'inégal palindrome. Neige. Bagatelle, dira Hercule. Le brut repentir, cet écrit né Pérec ...

et se termine par :

... S'il porte, sépulcral, ce repentir, cet écrit ne perturbe le lucre : Haridelle, ta gabegie ne mord ni la plage, ni l'écart.

Conversions

Ecrire une fonction *lettres()* qui prend comme paramètre un entier positif ou nul, et renvoie sa valeur sous forme de lettres, conformément à l'usage du français :

--> lettres(771)

ans = sept cent soixante et onze

--> lettres(791)

ans = sept cent quatre-vingt-onze

--> lettres(700)

ans = sept cents

Romain

```
function [z]=rom2dec(a)
z=0;
while length(a)>1 do
if part(a,1)=='M' then z=z+1000; a=part(a,2:length(a));
elseif part(a,[1 2])=='CM' then z=z+900; a=part(a,3:length(a));
elseif part(a,1)=='D' then z=z+500; a=part(a,2:length(a));
elseif part(a,[1 2])=='CD' then z=z+400; a=part(a,3:length(a));
elseif part(a,1)=='C' then z=z+100; a=part(a,2:length(a));
elseif part(a,[1 2])=='XC' then z=z+90; a=part(a,3:length(a));
elseif part(a,1)=='L' then z=z+50; a=part(a,2:length(a));
elseif part(a,[1 2])=='XL' then z=z+40; a=part(a,3:length(a));
elseif part(a,1)=='X' then z=z+10; a=part(a,2:length(a));
elseif part(a,[1 2])=='IX' then z=z+9; a=part(a,3:length(a));
elseif part(a,1)=='V' then z=z+5; a=part(a,2:length(a));
elseif part(a,[1 2])=='IV' then z=z+4; a=part(a,3:length(a));
elseif part(a,1)=='I' then z=z+1; a=part(a,2:length(a));
else error('Chaîne de caractères incompatible');
end;
end;
```

```
function [a]=dec2rom(n)
if n<0 | n>10000 then error('Valeur hors limites'); end
a="";
while n>0 do
if n>=1000 then n=n-1000; a=a+'M';
elseif n>=900 then n=n-900; a=a+'CM';
elseif n>=500 then n=n-500; a=a+'D';
elseif n>=400 then n=n-400; a=a+'CD';
elseif n>=100 then n=n-100; a=a+'C';
elseif n>=90 then n=n-90; a=a+'XC';
elseif n>=50 then n=n-50; a=a+'L';
elseif n>=40 then n=n-40; a=a+'XL';
elseif n>=10 then n=n-10; a=a+'X';
elseif n>=9 then n=n-9; a=a+'IX';
elseif n>=5 then n=n-5; a=a+'V';
elseif n>=4 then n=n-4; a=a+'IV';
elseif n>=1 then n=n-1; a=a+'I';
end
end
```

```
-->rom2dec('MDCIX')
```

```
ans = 1609.
```

```
-->dec2rom(1609)
```

```
ans = MDCIX
```

```
-->dec2rom(6893)
```

```
ans = MMMMMMDCCCXCIII
```

Palindrome

```
function [b]=palindrome(s)
b=%t;
for i=1:floor(length(s)/2) do
if part(s,i)<>part(s,length(s)-i+1) then b=%f; end
end
```

```
function [b]=caractere(s)
b=%t;
str='âââéèëêçîñôöüù .,:!?' ;
a=strindex(str,s);
if isempty(a)==%f then b=%f; end
```

```
function [z]=transformestr(a)
n=length(a);z="";
for i=1:n do
if caractere(part(a,i))==%t then z=z+part(a,i);
else
stra='âââ'; stre='éèëê'; strc='ç'; stri='ï'; stro='ôö'; stru='üùù';
p=0;
b=strindex(stra,part(a,i)); if isempty(b)==%f then p=1; end;
b=strindex(stre,part(a,i)); if isempty(b)==%f then p=2; end;
b=strindex(strc,part(a,i)); if isempty(b)==%f then p=3; end;
b=strindex(stri,part(a,i)); if isempty(b)==%f then p=4; end;
b=strindex(stro,part(a,i)); if isempty(b)==%f then p=5; end;
b=strindex(stru,part(a,i)); if isempty(b)==%f then p=6; end;
select p
case 1 then z=z+'a';
case 2 then z=z+'e';
case 3 then z=z+'c';
case 4 then z=z+'i';
case 5 then z=z+'o';
case 6 then z=z+'u';
end
```

```
end
end
z=code2str(-abs(str2code(z)))
```

```
function [b]=palindrome2(s)
s=transformestr(s);
b=%t;
for i=1:floor(length(s)/2) do
if part(s,i)<>part(s,length(s)-i+1) then b=%f; end
end
```

```
-->palindrome2('Esopo reste ici et roupille')
ans = F
-->palindrome2('Esopo reste ici et se repose')
ans = T
```