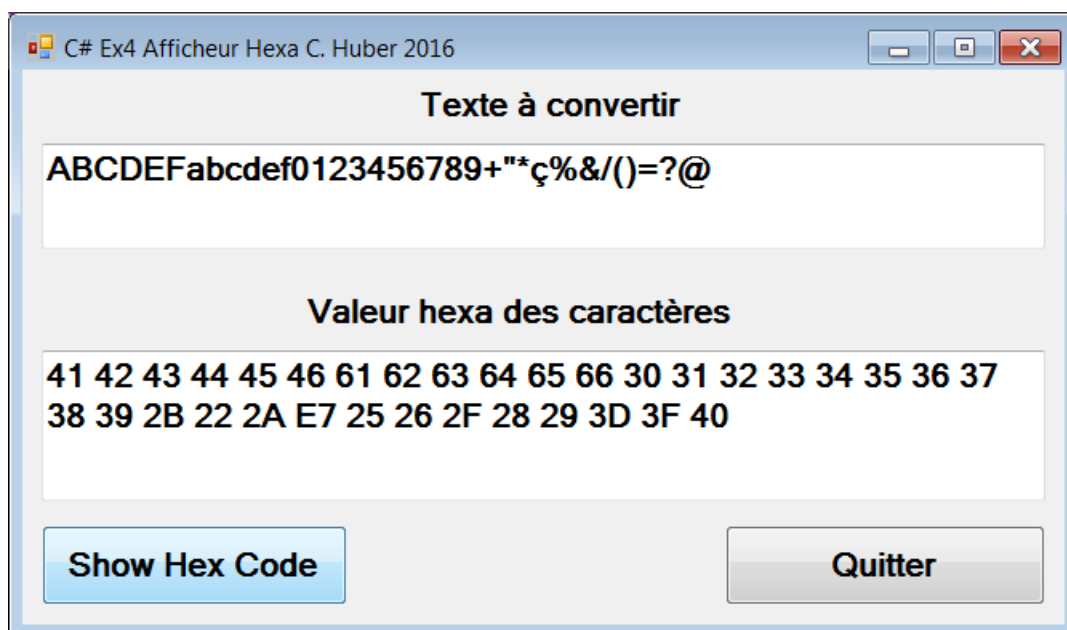


Exercice 4

AFFICHAGE DE LA VALEUR HEXADÉCIMAL DE CHACUN DES CARACTÈRES D'UNE CHAÎNE

Il s'agit de réaliser l'affichage en hexadécimal de la valeur du code ASCII de chacun des caractères du texte à convertir, lorsque que l'on active le bouton "Show hex code".

Voici un exemple de comment peut se présenter la boîte de dialogue en exécution.



La conversion est déclenchée par l'activation du bouton "Show hex code".
Le bouton "Quitter" doit fermer l'application.

CONTRAINTES POUR LA REALISATION

Le TextBox d'entrée doit être nommé txtInput, celui de sortie txtOutput. Les boutons de commande doivent être nommés btnShow et btnQuitter.

La feuille doit porter un titre (propriété Text), la feuille doit être renommée formEx4 (propriété (Name)) et enregistrée sous formEx4.cs.

ELEMENTS DE THEORIE

CONVERSION ENTIER EN STRING

Voici obtenu de <http://stackoverflow.com/questions/3081916/convert-int-to-string>
Les différentes façons de convertir un entier en string :

```
string s = i.ToString();
string s = Convert.ToString(i);
string s = string.Format("{0}", i);
string s = "" + i;
string s = string.Empty + i;
string s = new StringBuilder().Append(i).ToString();
```

Cela donne des pistes pour des recherches dans l'aide.

LA METHODE INT32.TOSTRING

La méthode `.ToString` convertit la valeur numérique de cette instance en sa représentation équivalente sous forme de chaîne. Cette méthode possède une liste de surcharge.

La variante: `public string ToString(string format)`

Permet grâce à la chaîne de formatage de choisir la forme du résultat. Voici les exemples obtenu de l'aide.

```
int value = -16325;
string specifier;

// Use standard numeric format specifier.
specifier = "G";
Console.WriteLine("{0}: {1}", specifier, value.ToString(specifier));
// Displays:    G: -16325

specifier = "C";
Console.WriteLine("{0}: {1}", specifier, value.ToString(specifier));
// Displays:    C: ($16,325.00)

specifier = "D8";
Console.WriteLine("{0}: {1}", specifier, value.ToString(specifier));
// Displays:    D8: -00016325

specifier = "E4";
Console.WriteLine("{0}: {1}", specifier, value.ToString(specifier));
// Displays:    E4: -1.6325E+004

specifier = "e3";
Console.WriteLine("{0}: {1}", specifier, value.ToString(specifier));
// Displays:    e3: -1.633e+004

specifier = "F";
Console.WriteLine("{0}: {1}", specifier, value.ToString(specifier));
// Displays:    F: -16325.00

specifier = "N";
Console.WriteLine("{0}: {1}", specifier, value.ToString(specifier));
// Displays:    N: -16,325.00
```

```

specifier = "P";
Console.WriteLine("{0}: {1}", specifier,
(value/100000).ToString(specifier));
// Displays:      P: -16.33 %

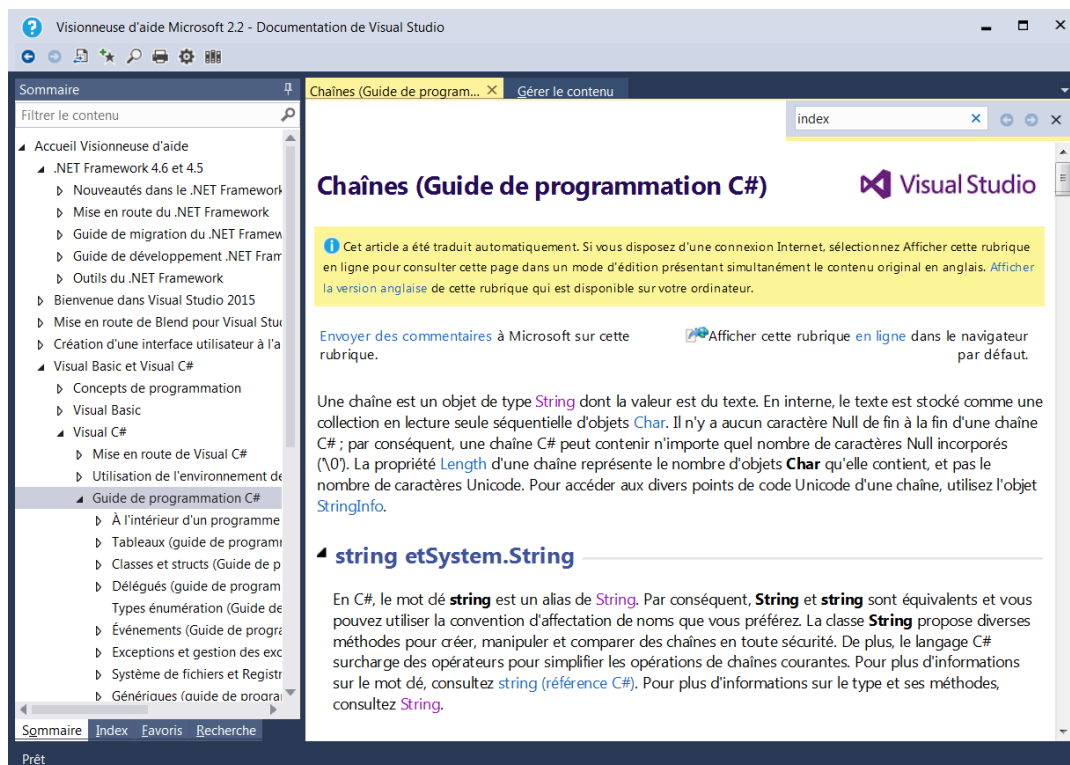
specifier = "X";
Console.WriteLine("{0}: {1}", specifier, value.ToString(specifier));
// Displays:      X: FFFFC03B

// Use custom numeric format specifiers.
specifier = "0,0.000";
Console.WriteLine("{0}: {1}", specifier, value.ToString(specifier));
// Displays:      0,0.000: -16,325.000

specifier = "#,#.00#;(#,#.00#)";
Console.WriteLine("{0}: {1}", specifier,
(value*-1).ToString(specifier));
// Displays:      #,#.00#;(#,#.00#): 16,325.00

```

AIDE SUR LES CHAINES



Dans cette section assez importante on trouve :

Accès à des caractères

Vous pouvez utiliser la notation de tableau avec une valeur d'index pour accéder en lecture seule aux différents caractères, comme dans l'exemple suivant :

```

JavaScript C# C++ F# JScript VB
Copier dans le Presse-papiers

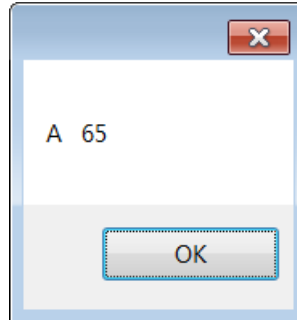
string s5 = "Printing backwards";

for (int i = 0; i < s5.Length; i++)
{
    System.Console.Write(s5[s5.Length - i - 1]);
}
// Output: "sdrawkcaB gnitniR"

```

On en déduit qu'il est possible d'obtenir un caractère de la chaîne en utilisant son indice.
par exemple :

```
string strIn = "ABCDE";  
char first = strIn[0];  
int code = first;  
string Mess = first.ToString() + "   " + code.ToString();  
MessageBox.Show(Mess);
```



Cet exemple permet de comprendre la nature du type char et le besoin de passer par un int pour obtenir le code ascii du caractère.

LA METHODE STRING.TOCHARARRAY

Copie les caractères de cette instance vers un tableau de caractères Unicode.

```
string delimStr = " , . : ";  
char [] delimiter = delimStr.ToCharArray();
```

CONCLUSION SUR L'AIDE

Les différents éléments de l'aide doivent permettre en les combinant de faciliter la réalisation de l'exercice.

TABLE DES CARACTERES ASCII (0 A 127)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	>	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

TABLE DES CARACTERES ASCII ETENDU

Uniquement le code en décimal et le symbole correspondant.

128	Ç	144	É	161	í	177	☐	193	±	209	〒	225	ß	241	±
129	ù	145	æ	162	ó	178	☐	194	⌈	210	⌈	226	Γ	242	≥
130	é	146	Æ	163	ú	179		195	⌋	211	⌋	227	π	243	≤
131	â	147	ô	164	ñ	180	†	196	—	212	⌋	228	Σ	244	∫
132	ä	148	ö	165	Ñ	181	‡	197	+	213	ƒ	229	σ	245	∫
133	à	149	ò	166	ª	182	‡	198	†	214	ƒ	230	μ	246	÷
134	â	150	û	167	º	183	¶	199	‡	215	‡	231	τ	247	≈
135	ç	151	ù	168	¿	184	¶	200	⌋	216	‡	232	Φ	248	°
136	ê	152	—	169	—	185	¶	201	ƒ	217	‡	233	Θ	249	.
137	ë	153	Ö	170	¬	186		202	⌋	218	‡	234	Ω	250	.
138	è	154	Û	171	½	187	¶	203	〒	219	■	235	δ	251	√
139	î	156	£	172	¾	188	¶	204	‡	220	■	236	∞	252	—
140	ï	157	¥	173	¡	189	¶	205	=	221	■	237	φ	253	²
141	ì	158	—	174	«	190	‡	206	‡	222	■	238	ε	254	■
142	Ä	159	ƒ	175	»	191	‡	207	±	223	■	239	∩	255	
143	Å	160	á	176	☐	192	⌋	208	⌋	224	α	240	≡		

Source: www.LookupTables.com