

PEEK ET POKE POUR LE HP-28

Après le PEEK de Wlodek Mier-Jedrzejowicz publié dans JPC 47 qui ne permettait que peu de facilités, il devenait nécessaire de créer un PEEK plus facile à utiliser. Grâce aux découvertes réalisées sur la structure des objets dans le HP-28C (voir l'article précédent) il est devenu possible de créer un PEEK fiable et commode.

Syntaxe :
 Niveau 2 : adresse du PEEK
 Niveau 1 : nombre de quartets à lire.

Ces deux nombres doivent être des entiers binaires.
 Attention : aucune vérification n'est faite sur le type des objets pour rendre le programme le plus court possible.

En sortie : une chaîne de caractères contenant les caractères 0, 1, ..., 9, A, ..., E, F représentant le contenu des quartets (nombres hexadécimaux).

Pour entrer le programme, entrez d'abord ASS et →LEX, si ce n'est pas déjà fait. Ensuite, entrez sous la forme d'une chaîne hexadécimale les nombres contenus dans la colonne *Code* du listing, en une seule ligne, sans espace, puis exécutez ASS et →LEX (voir article précédent et JPC 50 page 8). Stockez le résultat dans 'PEEK'. Ce programme est alors prêt à servir.

Code	Label	Mnemonic	Commentaires			
76C20		CON(5) #02C67	Début de programme	113	A=R3	
5A3C1		CON(5) #1C3A5	Vérifie la présence d'au moins deux éléments sur la pile.	133	AD1EX	
07A20		CON(5) #02A70	.	143	A=DAT1 A	
60000		CON(5) #00006	. #0	133	AD1EX	D1=^ BYTE (objet)
0		CON(1) #0	.	174	D1=D1+ 5	D1=^ BYTE (où écrire)
BF543		CON(5) #345FB	additionne 2 entiers hexadécimaux on recrée un nouveau nombre hexa en mémoire.	AE0	A=0	
5F020		CON(5) #020F5	SWAP	15A0	A=DAT0 1	On lit un quartet
07A20		CON(5) #02A70	.	3103	LCHEX #30	.
60000		CON(5) #00006	. #0	A6A	A=A+C B	. Contenu du
0		CON(1) #0	.	3193	LCHEX #39	. quartet ->
BF543		CON(5) #345FB	+	9EA	?A<=C B	. code-caractère
5F020		CON(5) #020F5	SWAP	90	GOYES L11	. ("1", "2"..., "F")
E4A20		CON(5) #02A4E	"" amorce de la chaîne qui contiendra le résultat du PEEK	3170	LCHEX #07	.
50000		CON(5) #00005		A6A	A=A+C B	.
FB920 L0		CON(5) #029BF	Byte 00 (sera utilisé pour créer un CHR)	149	L11 DAT1=A B	code dans octet
00		CON(2) #00		113	A=R3	On récupère D1
69C20		CON(5) #02C96	Assembly code	133	AD1EX	
C6000 deb1		CON(5) fin1-deb1	Longueur du code	112	A=R2	On récupère D0
132		AD0EX	R2 := D0	132	AD0EX	
102		R2=A		142	A=DAT0 A	.
133		AD1EX	R3 := D1	164	D0=D0+ 5	. fin de routine
103		R3=A		808C	PC=(A)	.
133		AD1EX		FC630 fin1	CON(5) #036CF	Cette routine prend une chaîne + un BYTE dans la pile et retourne Chaîne + CHR (BYTE)
17E		D1=D1+ 15	D1 = ^ adresse	69C20	CON(5) #02C96	Assembly code
143		A=DAT1 A		E6000 deb2	CON(5) fin2-deb2	Longueur
133		AD1EX		133	AD1EX	Sauvegarde de D1 dans R3
179		D1=D1+ 10		103	R3=A	
143		A=DAT1 A		133	AD1EX	
133		AD1EX		179	D1=D1+ 10	D1=^ où est l'adresse du PEEK
17E		D1=D1+ 15		143	A=DAT1 A	D1= Adresse du PEEK
143		A=DAT1 A		133	AD1EX	On l'incrémente
132		AD0EX		170	D1=D1+ 1	
102		R2=A		133	AD1EX	
133		AD1EX		141	DAT1=A A	et on l'écrit
179		D1=D1+ 10		113	A=R3	
143		A=DAT1 A		133	AD1EX	
132		AD0EX		174	D1=D1+ 5	
102		R2=A		143	A=DAT1 A	
133		AD1EX		133	AD1EX	
179		D1=D1+ 10		179	D1=D1+ 10	D1=^ où est la longueur A= longueur PEEK
143		A=DAT1 A		143	A=DAT1 A	
132		AD0EX		133	AD1EX	
102		R2=A		100	D1=D1- 1	On décrémente longueur
133		AD1EX		133	AD1EX	
179		D1=D1+ 10		141	DAT1=A A	
143		A=DAT1 A		8A8	?A=0 A	Longueur nulle -> fini
132		AD0EX		11	GOYES L21	
102		R2=A		132	AD0EX	On décrémente D0 pour continuer en L0 lors du prochain A=DAT0 D0=D0+ 5 PC=(A)
133		AD1EX		340F000	LCHEX #000F0	
179		D1=D1+ 10		FA	A=A-C A	
143		A=DAT1 A		132	AD0EX	
132		AD0EX		113	L21 A=R3	

```

133      AD1EX          On récupère D1
142      A=DAT0 A      .
164      D0=D0+ 5     . fin de routine
808C     PC=(A)        .
69C20    CON(5) #02C96
72000 deb3 CON(5) fin3-deb3
133      AD1EX          . dans la pile
103      R3=A          . 3 = #adr
133      AD1EX          . 2 = #len
143      A=DAT1 A      . 1 = "résultat"
179      D1=D1+ 10    . devient
141      DAT1=A A      . 3 = "résultat"
113      A=R3          . 2 = #len
133      AD1EX          . 1 = "résultat"
142      A=DAT0 A      .
164      D0=D0+ 5     . après le DROP2
808C     PC=(A)        . 1 = "résultat"
A2120 fin3 CON(5) #0212A DROP2
09F20    CON(5) #02F90 fin de structure

```

Après le PEEK voici le POKE :

Syntaxe :

Niveau 2 : adresse (entier binaire)

Niveau 1 : chaîne contenant les codes.

Exemple : #40078 "FFFFFFF" POKE place 15 (#F en hexadécimal) dans les 8 quartets #40078, #40079,...,#4007F.

Entrée du programme : taper les nombres de la première colonne sous forme d'une chaîne, en une seule ligne, sans espace puis [ASS] et [-LEX].

Code	Label	Mnemonic	Commentaires
76C20		CON(5) #02C67	Structure programme
5A3C1		CON(5) #1C3A5	2 éléments dans la pile
69C20		CON(5) #02C96	Assembly code
C9000	deb	CON(5) fin-deb	Longueur code
132		AD0EX	
103		R3=A	On sauve D0
AFC		ABEX W	
102		R2=A	On sauve B
133		AD1EX	
101		R1=A	On sauve D1
133		AD1EX	
174		D1=D1+ 5	D1 pointe sur #adresse dans la pile
143		A=DAT1 A	
133		AD1EX	
179		D1=D1+ 10	
143		A=DAT1 A	
132		AD0EX	D0= ^ du POKE
111		A=R1	
133		AD1EX	
143		A=DAT1 A	

```

133      AD1EX
174      D1=D1+ 5
143      A=DAT1 A
3450000  LCHEX #00005
DC       ABEX A      B= longueur + 5
174      D1=D1+ 5      D1= ^ début chaîne
E1      L1 B=B-C A
8A9     ?B=0 A      POKE fini ?
13      GOYES L3
14B     A=DAT1 B
3103    LCHEX #30
B6A     A=A-C B      . code ASCII
3190    LCHEX #09      . -> de 0 à F
9EA     ?A<=C B
90      GOYES L2
3170    LCHEX #07
B6A     A=A-C B
1580    L2 DAT0=A 1    On fait le POKE
160     D0=D0+ 1      . caractère suivant
171     D1=D1+ 2
3420000 LCHEX #00002
6DCF    GOTO L1
112     L3 A=R2
AFC     ABEX W      On récupère B
113     A=R3
132     AD0EX      On récupère D0
111     A=R1
133     AD1EX      On récupère D1
142     A=DAT0 A
164     D0=D0+ 5   . fin de routine
808C    PC=(A)
A2120   CON(5) #0212A DROP2
09F20   CON(5) #02F90 fin de structure

```

Remarque : le programme sauve les éléments présents dans la pile avec la routine #1C3A5. Par la suite, un appel est fait à la routine #0212A qui est un DROP2 différent du standard, parce qu'il ne sauvegarde pas dans LAST. Ainsi, la modification faite par PEEK dans la pile n'est pas répercutée dans LAST.

Paul Courbis (392)

