

Suite et fin de la commande LINPUT

gestion flèche vers le bas

EDDE- 18 CLC prépare une addition
 EDDF- 8A TXA copie dans A le nombre de caractères saisis
 EDE0- 65 F5 ADC F5 et ajoute nombre de colonnes actives selon 026A
 EDE2- B0 18 BCS EDFC si résultat > 255, branche en EDFC avec Y = 4
 EDE4- C5 F2 CMP F2 teste si résultat >= nombre de caract requis
 EDE6- B0 14 BCS EDFC si oui, continue en EDFC avec Y = 4
 EDE8- AA TAX le nbre de caract saisis augmente de 38 (ou 40)
 EDE9- A9 0A LDA #0A "flèche vers le bas"
 EDEB- D0 A3 BNE ED90 aff caract de CTRL valide et reboucle en saisie

Suite "4" de l'analyse des codes de contrôle

EDED- C8 INY incrémente Y (qui passe à 5)
 EDEE- C9 0B CMP #0B est-ce "**flèche vers le haut**"?
 EDF0- D0 A1 BNE ED93 si non, reboucle en ED64 (saisie au clavier)

Gestion flèche vers le haut

EDF2- 8A TXA copie dans A le nombre de caractères saisis
 EDF3- E5 F5 SBC F5 et retire nombre de colonnes actives selon 026A
 EDF5- 90 05 BCC EDFC si résultat < 0, continue en EDFC avec Y = 5
 EDF7- AA TAX le nbre de caract saisis diminue de 38 (ou 40)
 EDF8- A9 0B LDA #0B "flèche vers le haut"
 EDFA- D0 94 BNE ED90 aff caract de CTRL valide et reboucle en saisie

Sortie

EDFC- C0 02 CPY #02 Y vaut-il 0 ou 1? (RETURN ou ESC)
 EDFE- 90 04 BCC EE04 si oui, saute les deux instructions suivantes
 EE00- A5 F3 LDA F3 teste si S (permet de sortir avec les flèches)
 EE02- 30 8F BMI ED93 si oui, reboucle en ED64 (saisie au clavier)
EE04- 84 F4 STY F4 F4 contient désormais le mode de sortie: 0 si RETURN, 1 si ESC, 2 si flèche gauche, 3 si flèche droite, 4 si flèche vers le bas, 5 si flèche vers le haut et 6 si sortie auto en fin de fenêtre.

Positionne le curseur à la fin de la fenêtre

EE06- 20 40 D7 JSR D740 sous-programme (s/p) "CURSEUR OFF"
EE09- E8 INX incrémente le nombre de caractères
 EE0A- E4 F2 CPX F2 le nombre requis est-il atteint?
 EE0C- B0 05 BCS EE13 si X >= F2, continue en EE13
 EE0E- 20 76 EE JSR EE76 affiche "flèche droite"
 EE11- D0 F6 BNE EE09 si X < F2 reboucle en EE09

Recopie la fenêtre dans la variable alphanumérique

EE13- A5 F2 LDA F2 nombre de caractères requis
 EE15- 20 64 D2 JSR D264 Réserv chaîne lg A, sauv lg D0 et adr D1/D2
 EE18- A4 F2 LDY F2
EE1A- 84 F5 STY F5 nombre de caractères requis
 EE1C- AC 69 02 LDY 0269 n° de colonne du curseur (de 0 à 39)
 EE1F- B1 12 LDA (12),Y lit caractère sous le curseur
 EE21- C9 20 CMP #20 teste si >= #20 (caractère proprement dit)
 EE23- B0 02 BCS EE27 si oui, saute l'instruction suivante
 EE25- 09 80 ORA #80 si non, force à 1 le b7 du code lu au curseur, afin de retomber sur des caractères affichables normalement. A l'écran les attributs vidéo sont codés de 0 (encre noire) à 31 (GRAPHICS 50Hz). Mais, pour être inclus dans une chaîne, ces attributs sont codés de 128 à 159.

EE27- A4 F5 LDY F5 nombre de caractères requis
 EE29- 88 DEY que l'on diminue
 EE2A- 08 PHP Sauvegarde les indicateurs 6502 dont Z
 EE2B- 91 D1 STA (D1),Y écrit le caract ou code lu dans la chaîne
 EE2D- 20 73 EE JSR EE73 affiche "flèche gauche" (recule le curseur)
 EE30- 28 PLP Récupère les indicateurs 6502 dont Z
 EE31- D0 E7 BNE EE1A reboucle tant qu'il en reste à copier

Retourne à la fin de la fenêtre

EE33- A6 F2 LDX F2 nombre de caractères requis
EE35- 20 76 EE JSR EE76 affiche "flèche droite"
 EE38- CA DEX décrémente le nombre de caractères requis
 EE39- D0 FA BNE EE35 reboucle tant que la fin n'est pas atteinte

Remplace les caractères de remplissage par des espaces

EE3B- 06 F3 ASL F3 les bits indiquant les paramètres sélectionnés
EE3D- 06 F3 ASL F3 sont décalés 2 fois à gauche (J passe en b7, K en
b6 et E en b5), car S et C ont déjà été traités et peuvent être perdus.
EE3F- A4 F2 LDY F2 nombre de caractères requis
EE41- 88 DEY que l'on diminue
EE42- B1 D1 LDA (D1),Y lit caract de chaîne en commençant par la fin
EE44- CD 75 C0 CMP C075 est-ce un caractère de remplissage?
EE47- D0 18 BNE EE61 si non, "justification" finie, continue en EE61
EE49- A9 20 LDA #20 si oui, remplace le caractère lu par un espace
EE4B- 24 F3 BIT F3 teste si **J** ou **K** était demandé (**J** pour justifier
la variable alphanum en ajoutant des espaces, sans affecter l'écran et **K** pour
justifier à l'écran seulement, sans affecter la variable alphanum).
EE4D- 10 02 BPL EE51 si **J** non demandé, saute l'instruction suivante
EE4F- 91 D1 STA (D1),Y écrit un espace à la place du caract de rempl
EE51- 50 06 BVC EE59 si **K** non demandé, saute les 2 instructions suiv
EE53- 20 2A D6 JSR D62A XAFCAR Affiche cet espace (et déplace à droite)
EE56- 20 73 EE JSR EE73 puis affiche "flèche gauche" (revient en place)
EE59- 20 73 EE JSR EE73 affiche "flèche gauche" (caractère précédent)
EE5C- 98 TYA teste s'il reste des caractères à examiner
EE5D- D0 E2 BNE EE41 si oui, reboucle en EE41
EE5F- 24 C8 BIT C8 si non, "justification" finie, continue en EE61

Repositionne en fin de fenêtre

EE60- C8 INY point de rebouclage: caractère suivant
EE61- 20 76 EE JSR EE76 affiche "flèche droite"
EE64- C4 F2 CPY F2 tous les caractères ont-ils été examinés?
EE66- D0 F8 BNE EE60 si non, reboucle en EE60
EE68- 60 RTS si oui, retourne à la commande LINPUT. Il eut été
préférable de récupérer les coordonnées d'origine et de repositionner
convenablement le curseur au début de la fenêtre en simulat un affichage.

Affiche X fois "flèche gauche"

EE69- 8A TXA teste le nombre de caractères à saisir
EE6A- F0 06 BEQ EE72 simple RTS si 0
EE6C- 20 73 EE JSR EE73 affiche "flèche gauche"
EE6F- CA DEX décrémente le nombre de caractères à saisir
EE70- D0 F7 BNE EE69 reboucle en EE69 tant qu'il en reste à saisir
EE72- 60 RTS

Affiche une "flèche gauche"

EE73- A9 08 LDA #08 A = "flèche gauche"
EE75- 2C A9 09 BIT 09A9 et continue en EE7A

Affiche une "flèche droite"

EE76- A9 09 LDA #09 A = "flèche droite"
EE78- 24 68 BIT 68 et continue en EE7A

Passe la marge en mode 38 colonnes

EE79- 68 PLA reprend dans A la flèche qui est sur pile
Affiche une "flèche gauche" ou une "flèche droite"

EE7A- 48 PHA empile "flèche gauche" ou "flèche droite"
EE7B- 20 2A D6 JSR D62A affiche "flèche gauche" ou "flèche droite"
EE7E- AD 6A 02 LDA 026A mode console (ici la flèche de A est perdue)
EE81- 29 20 AND #20 teste b5 (à 0, si mode normal 38 colonnes)
EE83- D0 07 BNE EE8C continue en EE8C si mode 40 colonnes
EE85- AD 69 02 LDA 0269 n° de colonne (abscisse x)
EE88- 29 FE AND #FE ET 1111 1110 donne 0 si n° colonne vaut 0 ou 1
EE8A- F0 ED BEQ EE79 branche en EE79 si dans la marge
EE8C- 68 PLA retire "flèche gauche" ou "flèche droite"
EE8D- 60 RTS de la pile et retourne

Copie la longueur et l'adresse de la chaîne en B8/B9/BA

EE8E- A0 02 LDY #02 pour 3 copies d'octets (longueur et adr chaîne)
EE90- B9 D0 00 LDA 00D0,Y lecture octet, en commençant par la fin
EE93- 91 B8 STA (B8),Y écriture en B8 (longueur) et B9/BA (adresse)
EE95- 88 DEY vise octet précédent
EE96- 10 F8 BPL EE90 reboucle tant que Y est supérieur ou égal à 0
EE98- 60 RTS

Si vous avez des questions concernant le travail déjà publié ou des souhaits
concernant les prochaines routines que nous pourrions examiner, n'hésitez pas
à nous écrire. André Chéramy, 54, rue de Sours, 28000 CHARTRES et Yann
Legrand, 608, rue de l'Eglise, "Les Templiers", 62610 LANDRETHUN LES ARDRES