

Institut d'enseignement à distance  
Licence 1 - Programmation impérative  
Documentation utilisateur - Cx25

BLANCHARD Patrice  
Numéro étudiant : 18904701

14 mai 2020

# Table des matières

1	Présentation du programme cx25.1	3
2	Comment exécuter le programme cx25.1 ?	7
3	Quelles sont les options du débogueur ?	7
4	Quelles erreurs peut-on rencontrer ?	12

# 1 Présentation du programme cx25.1

Ce programme n'est autre qu'un interprète du langage suivant :

add #	20	$A \leftarrow A + V$
add @	60	$A \leftarrow A + (@)$
add *@	E0	$A \leftarrow A + *(@)$
sub #	21	$A \leftarrow A - V$
sub @	61	$A \leftarrow A - (@)$
sub *@	E1	$A \leftarrow A - *(@)$
nand #	22	$A \leftarrow \neg[A \& V]$
nand @	62	$A \leftarrow \neg[A \& (@)]$
nand *@	E2	$A \leftarrow \neg[A \& *(@)]$
load #	00	$A \leftarrow V$
load @	40	$A \leftarrow (@)$
load *@	C0	$A \leftarrow *(@)$
store @	48	$(@) \leftarrow A$
store *@	C8	$*(@) \leftarrow A$
in @	49	$(@) \leftarrow \text{Entrée}$
in * @	C9	$*(@) \leftarrow \text{Entrée}$
out @	41	$\text{Sortie} \leftarrow (@)$
out *@	C1	$\text{Sortie} \leftarrow *(@)$
jump @	10	$PC \leftarrow @$
brn @	11	Si $A < 0$ : $PC \leftarrow @$
brnz @	12	si $A = 0$ : $PC \leftarrow @$

autrement dit, ce programme lit le code d'un autre programme présent dans un fichier, puis simule ce que un processeur fictif ferait pour exécuter ce code.

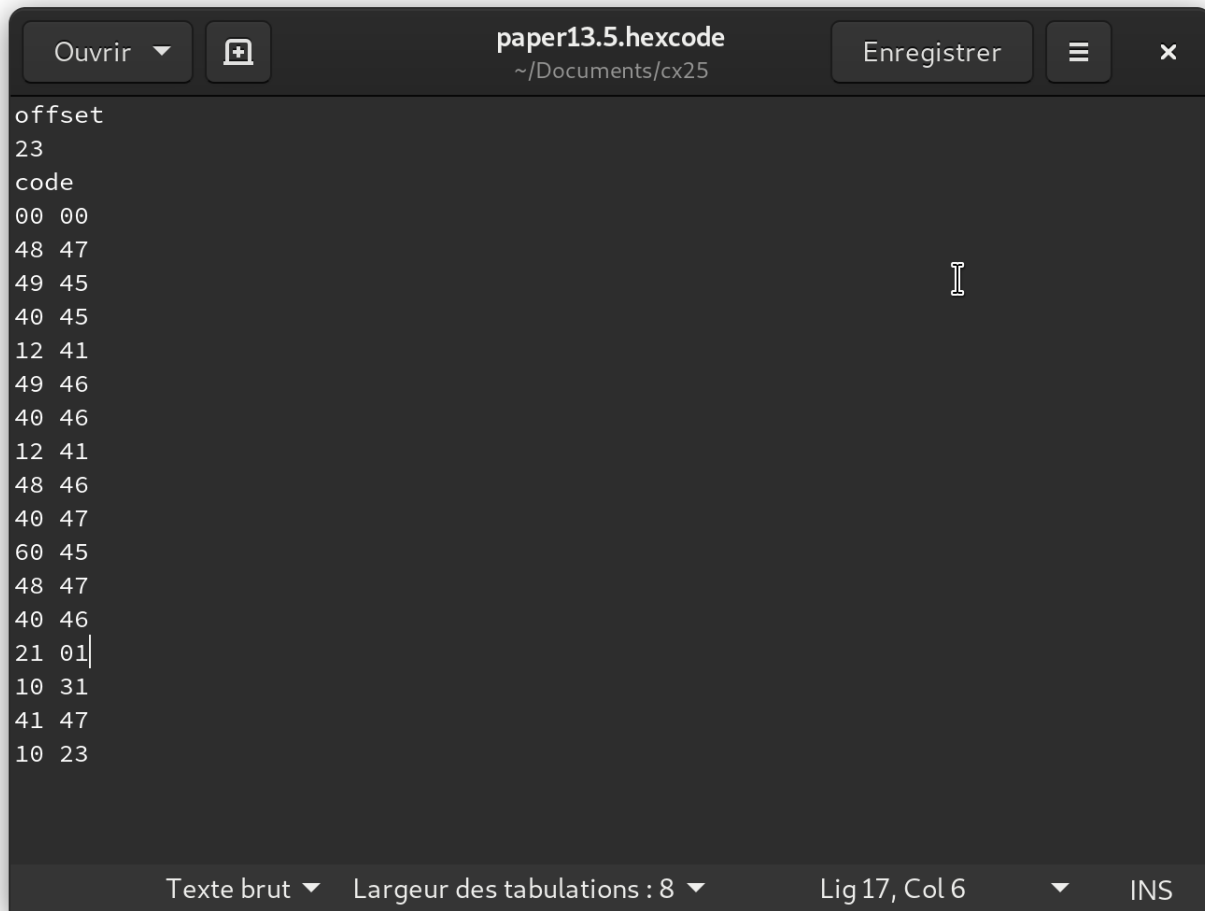
Le fichier comprend une suite d'instructions à effectuer (ou exécuter). L'adresse de la première instruction est placé dans le registre (Program Counter) qui sera incrémenter au fur et à mesure des cycles de décodage.

En effet, le programme répète un cycle constitué de trois phases : La première phase est la recherche d'instruction (Registre PC), la deuxième phase contient le décodage de l'instruction et sa réalisation proprement dite (recherche de l'opérande et calcul éventuel), les résultats des opérations sont toujours rendus dans l'accumulateur (registre A), la troisième phase permet de pointer vers l'instruction suivante.

## Un exemple d'utilisation de programme : le produit de deux nombres

ADD	CODE	MNÉMONIQUE	COMMENTAIRE
23	00	LOAD #00	CHARGER LE REGISTRE DE L'ACCUMULATEUR AVEC LA VALEUR 0
24	00		
25	48	STORE 47	RANGER LA VALEUR DU REGISTRE DE L'ACCUMULATEUR À L'ADRESSE 47 (RÉSULTAT)
26	47		
27	49	IN 45	SAISIR LA VALEUR DU PREMIER NOMBRE (NB1), L'ENREGISTRER À L'ADRESSE 45
28	45		
29	40	LOAD 45	CHARGER LE REGISTRE DE L'ACCUMULATEUR AVEC LA VALEUR DE NB1
2A	45		
2B	12	BRZ 41	SI NB1 = 0 ALORS PC ← 41 : AFFICHAGE DE LA VALEUR À L'ADRESSE 47 (RÉSULTAT)
2C	41		
2D	49	IN 46	SAISIR LA VALEUR DU SECOND NOMBRE : NB2
2E	46		
2F	40	LOAD 46	CHARGER LE REGISTRE DE L'ACCUMULATEUR AVEC LA VALEUR DE L'ADRESSE 46 : NB2
30	46		
31	12	BRZ 41	SI NB2 = 0 ALORS PC ← 41 : AFFICHAGE DU RÉSULTAT PRÉSENT À L'ADRESSE 47
32	41		
33	48	STORE 46	RANGER LA VALEUR DU REGISTRE DE L'ACCUMULATEUR À L'ADRESSE 46 (NB2)
34	46		
35	40	LOAD 47	CHARGER LE REGISTRE DE L'ACCUMULATEUR AVEC LA VALEUR DE L'ADRESSE 47 (RÉSULTAT)
36	47		
37	60	ADD 45	AJOUTER NB1 AU RÉSULTAT
38	45		
39	48	STORE 47	RANGER LA VALEUR DU REGISTRE DE L'ACCUMULATEUR À L'ADRESSE 47
3A	47		
3B	40	LOAD 46	CHARGER LA VALEUR DE L'ADRESSE 46 (NB2)
3C	46		
3D	21	SUB # 1	SOUSTRAIRE UN À LA VALEUR DE NB2 PRÉSENT DANS LE REGISTRE DE L'ACCUMULATEUR
3E	01		
3F	10	JUMP 31	RÉALISER UN SAUT À L'ADRESSE 31
40	31		
41	41	OUT 47	AFFICHER LA VALEUR DE L'ADRESSE 47 (RÉSULTAT)
42	47		
43	10	JUMP 23	RÉALISER UN SAUT À L'ADRESSE 23 (DÉBUT DU PROGRAMME)
44	23		
45	??		LE PREMIER NOMBRE : NB1
46	??		LE DEUXIÈME NOMBRE : NB2
47	??		LE RÉSULTAT FINAL OU TEMPORAIRE

Ce même programme sous la forme d'une suite numérique d'instructions :



The screenshot shows a hex editor window titled "paper13.5.hexcode" with the file path "~/Documents/cx25". The window contains a list of instructions in hexadecimal and decimal format. The instructions are as follows:

offset	hex	dec
23		
code		
00	00	00
48	47	47
49	45	45
40	45	45
12	41	41
49	46	46
40	46	46
12	41	41
48	46	46
40	47	47
60	45	45
48	47	47
40	46	46
21	01	01
10	31	31
41	47	47
10	23	23

The status bar at the bottom indicates "Texte brut", "Largeur des tabulations : 8", "Lig 17, Col 6", and "INS".

FIGURE 1 – Le programme d'une multiplication

```
patrice@g3-3590:~/Documents/cx25
(base) [patrice@g3-3590 cx25]$ ./cx25.1 paper13.5.hexcode
gdb - cx25.1
Voulez-vous afficher la valeur de PC?
(Continuer : O/N): o
Voulez-vous afficher les la valeur de A?
(Continuer : O/N): o

Début du programme :

PC : 035 | A : 000 | LOAD #00    A = 00
PC : 037 | A : 000 | STORE 47    data[47] = A
PC : 039 | A : 000 | IN 45      data[45] = input(val ?)
Saisir une valeur en décimal : 3

PC : 041 | A : 000 | LOAD 45    A = data[45]
PC : 043 | A : 003 | BRZ 41     Si A = 0 alors PC = 41
PC : 045 | A : 003 | IN 46      data[46] = input(val ?)
Saisir une valeur en décimal : 2

PC : 047 | A : 003 | LOAD 46    A = data[46]
PC : 049 | A : 002 | BRZ 41     Si A = 0 alors PC = 41
PC : 051 | A : 002 | STORE 46    data[46] = A
PC : 053 | A : 002 | LOAD 47    A = data[47]
PC : 055 | A : 000 | ADD 45     A += data[45]
PC : 057 | A : 003 | STORE 47    data[47] = A
PC : 059 | A : 003 | LOAD 46    A = data[46]
PC : 061 | A : 002 | SUB #01    A -= 01
PC : 063 | A : 001 | JUMP 31    PC = 31
PC : 049 | A : 001 | BRZ 41     Si A = 0 alors PC = 41
PC : 051 | A : 001 | STORE 46    data[46] = A
PC : 053 | A : 001 | LOAD 47    A = data[47]
PC : 055 | A : 003 | ADD 45     A += data[45]
PC : 057 | A : 006 | STORE 47    data[47] = A
PC : 059 | A : 006 | LOAD 46    A = data[46]
PC : 061 | A : 001 | SUB #01    A -= 01
PC : 063 | A : 000 | JUMP 31    PC = 31
PC : 049 | A : 000 | BRZ 41     Si A = 0 alors PC = 41
PC : 065 | A : 000 | OUT 47     print(data[47])
OUT en décimal : 6
(Continuer : O/N): n
(base) [patrice@g3-3590 cx25]$
```

FIGURE 2 – L'exécution de la multiplication par l'interprète

## 2 Comment exécuter le programme cx25.1 ?

De deux manières :

— `cx25.1 <nom_fichier>`

L'affichage de l'ensemble des instructions se produit sans attendre.

— `cx25.1 -d <nom_fichier>` active un débogueur (stepper)

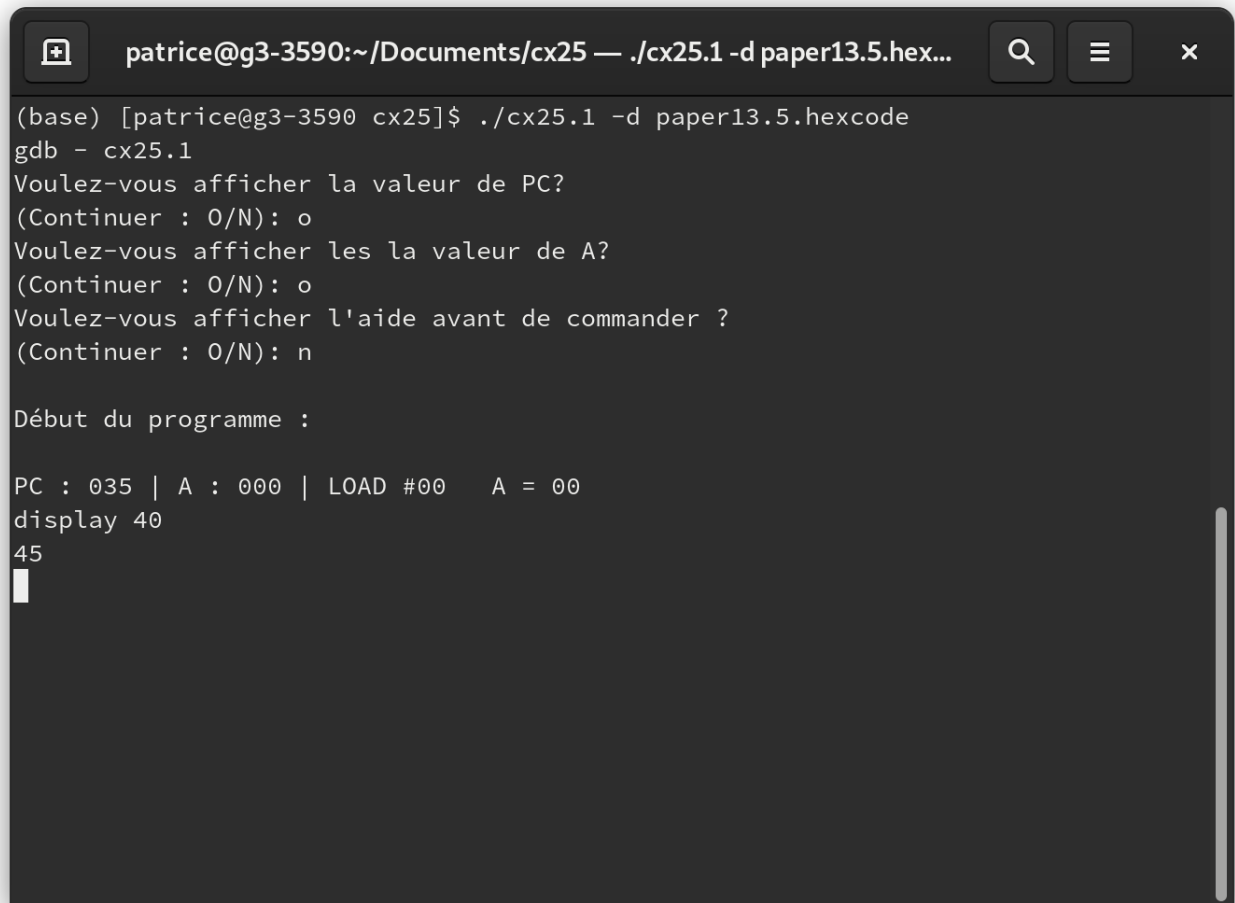
L'affichage des instructions se déroule pas à pas.

Dans les deux cas, au début du programme, il est possible d'afficher ou non : la valeur de l'instruction présente dans le Program Counter : PC, la valeur présente dans l'accumulateur : A.

Lorsque le débogueur est activé, une option d'aide est disponible au démarrage ou lors de l'exécution du programme renseignant des différentes commandes pouvant être mise en œuvre avant la fermeture de celui-ci.

## 3 Quelles sont les options du débogueur ?

`display <adresse>` : affiche la valeur de l'adresse demandée



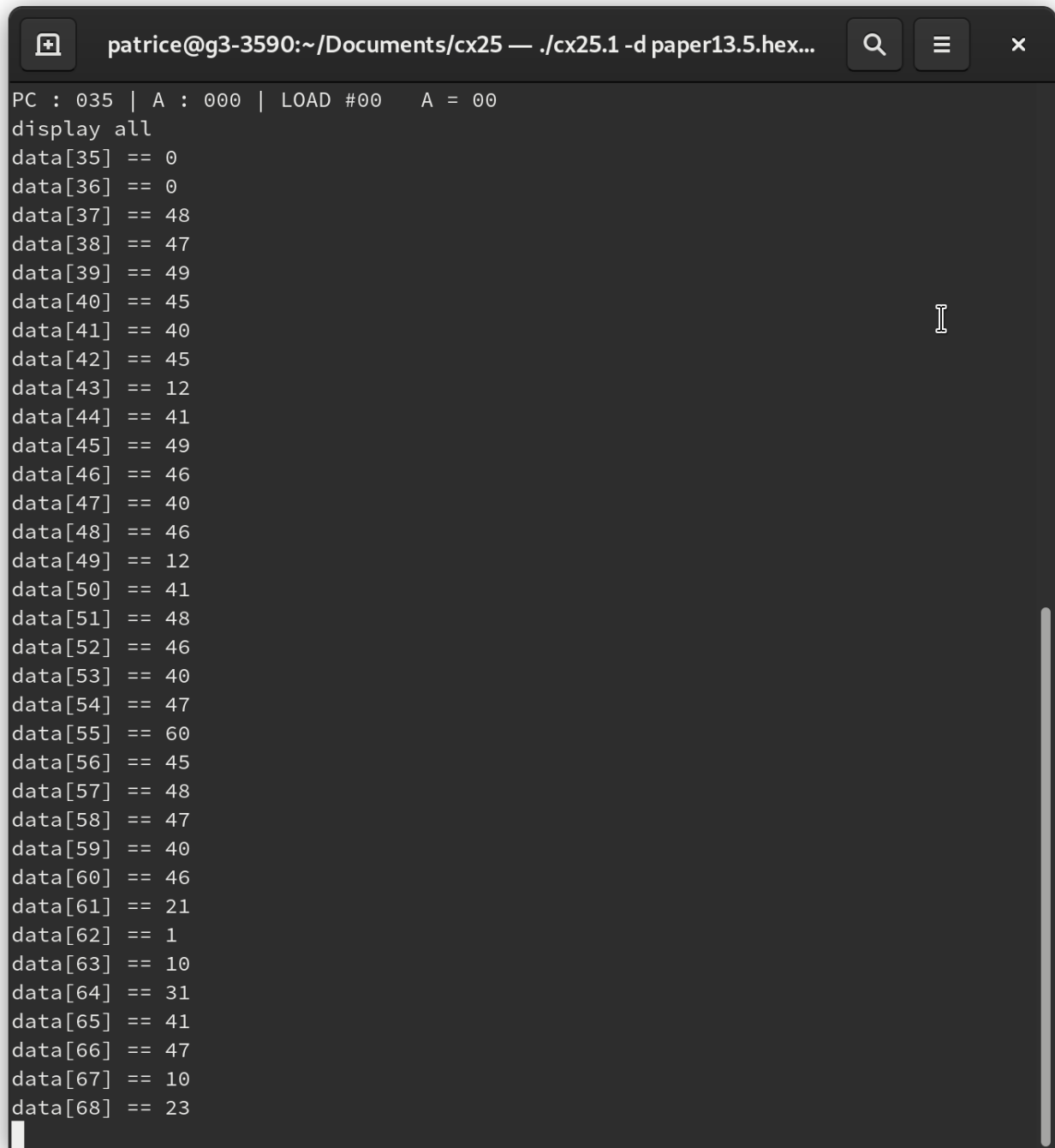
```
(base) [patrice@g3-3590 cx25]$ ./cx25.1 -d paper13.5.hexcode
gdb - cx25.1
Voulez-vous afficher la valeur de PC?
(Continuer : O/N): o
Voulez-vous afficher les la valeur de A?
(Continuer : O/N): o
Voulez-vous afficher l'aide avant de commander ?
(Continuer : O/N): n

Début du programme :

PC : 035 | A : 000 | LOAD #00   A = 00
display 40
45
█
```

FIGURE 3 – Un exemple d'utilisation de la commande `display <adresse>`

display all : affiche l'ensemble du programme provenant du fichier



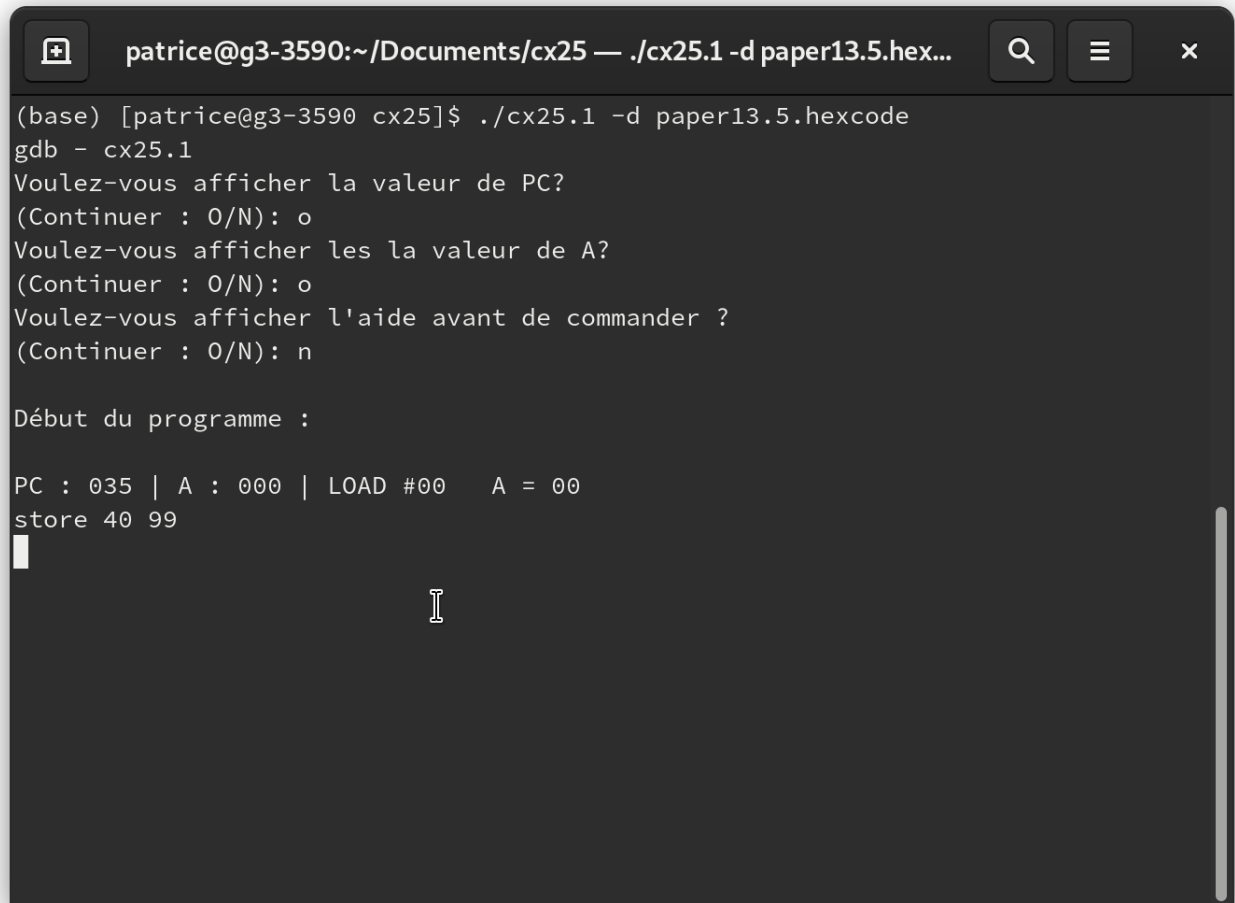
The screenshot shows a debugger window with a dark theme. The title bar indicates the user is patrice@g3-3590 and the current file is ~/Documents/cx25. The command line shows the debugger is running ./cx25.1 -d paper13.5.hex... The main window displays the output of the 'display all' command, showing the current PC value (035), register A (000), and the load number (00). Below this, a list of memory addresses and their corresponding values is shown, starting from data[35] and ending at data[68].

```
PC : 035 | A : 000 | LOAD #00    A = 00
display all
data[35] == 0
data[36] == 0
data[37] == 48
data[38] == 47
data[39] == 49
data[40] == 45
data[41] == 40
data[42] == 45
data[43] == 12
data[44] == 41
data[45] == 49
data[46] == 46
data[47] == 40
data[48] == 46
data[49] == 12
data[50] == 41
data[51] == 48
data[52] == 46
data[53] == 40
data[54] == 47
data[55] == 60
data[56] == 45
data[57] == 48
data[58] == 47
data[59] == 40
data[60] == 46
data[61] == 21
data[62] == 1
data[63] == 10
data[64] == 31
data[65] == 41
data[66] == 47
data[67] == 10
data[68] == 23
```

FIGURE 4 – Un exemple d'utilisation de la commande display all



store <adresse> <valeur> : enregistre une nouvelle valeur à l'adresse demandée, un fichier new\_version sera créé également. Il s'agit d'une copie du fichier initial avec la nouvelle valeur modifiée.



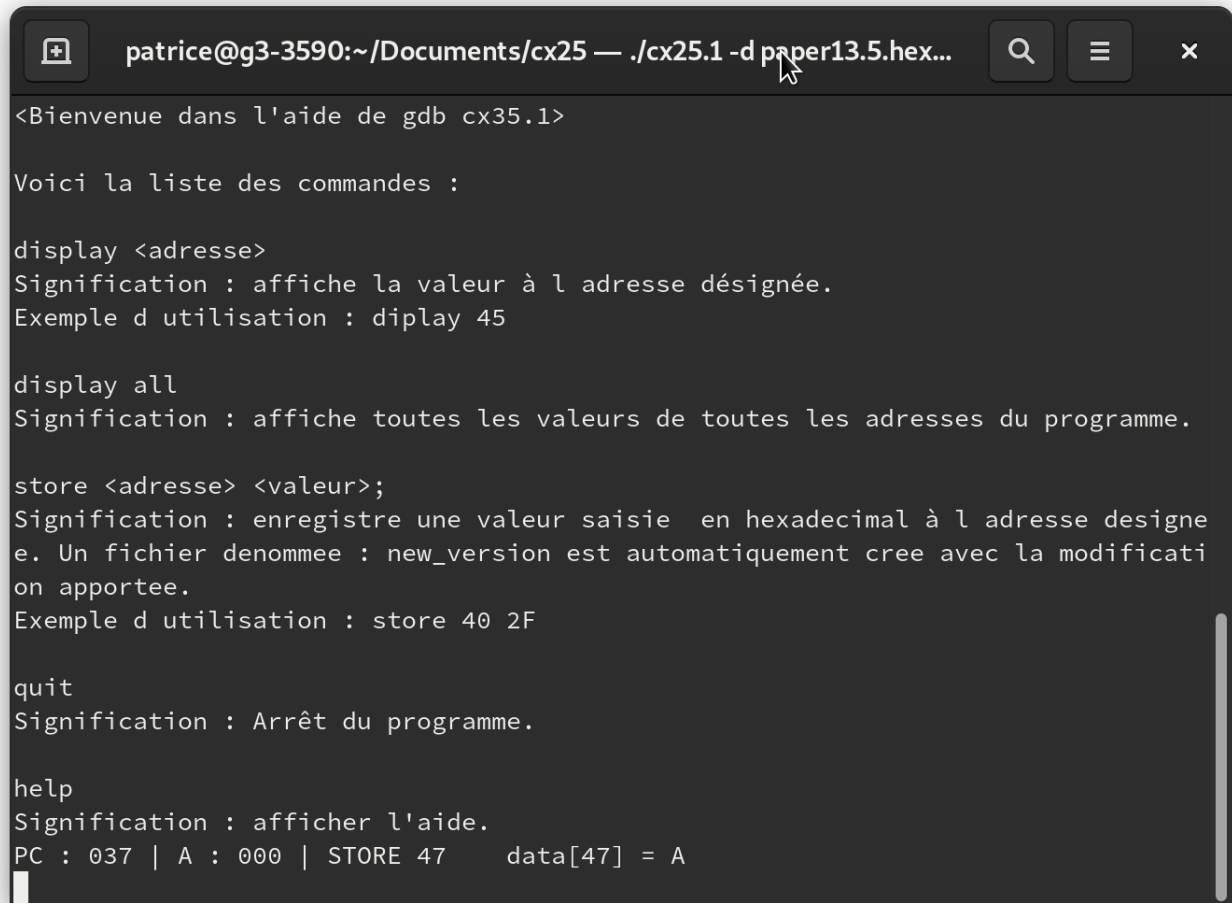
```
patrice@g3-3590:~/Documents/cx25 — ./cx25.1 -d paper13.5.hex...
(base) [patrice@g3-3590 cx25]$ ./cx25.1 -d paper13.5.hexcode
gdb - cx25.1
Voulez-vous afficher la valeur de PC?
(Continuer : O/N): o
Voulez-vous afficher les la valeur de A?
(Continuer : O/N): o
Voulez-vous afficher l'aide avant de commander ?
(Continuer : O/N): n

Début du programme :

PC : 035 | A : 000 | LOAD #00    A = 00
store 40 99
█
```

FIGURE 5 – Un exemple d'utilisation de la commande store <adresse> <valeur>

help : afficher l'aide



The screenshot shows a terminal window with a dark background. The title bar at the top reads "patrice@g3-3590:~/Documents/cx25 — ./cx25.1 -d paper13.5.hex...". The terminal content displays the output of the "help" command in GDB, listing various commands and their functions. At the bottom, the current state of the program is shown: "PC : 037 | A : 000 | STORE 47 data[47] = A".

```
<Bienvenue dans l'aide de gdb cx35.1>

Voici la liste des commandes :

display <adresse>
Signification : affiche la valeur à l adresse désignée.
Exemple d utilisation : diplay 45

display all
Signification : affiche toutes les valeurs de toutes les adresses du programme.

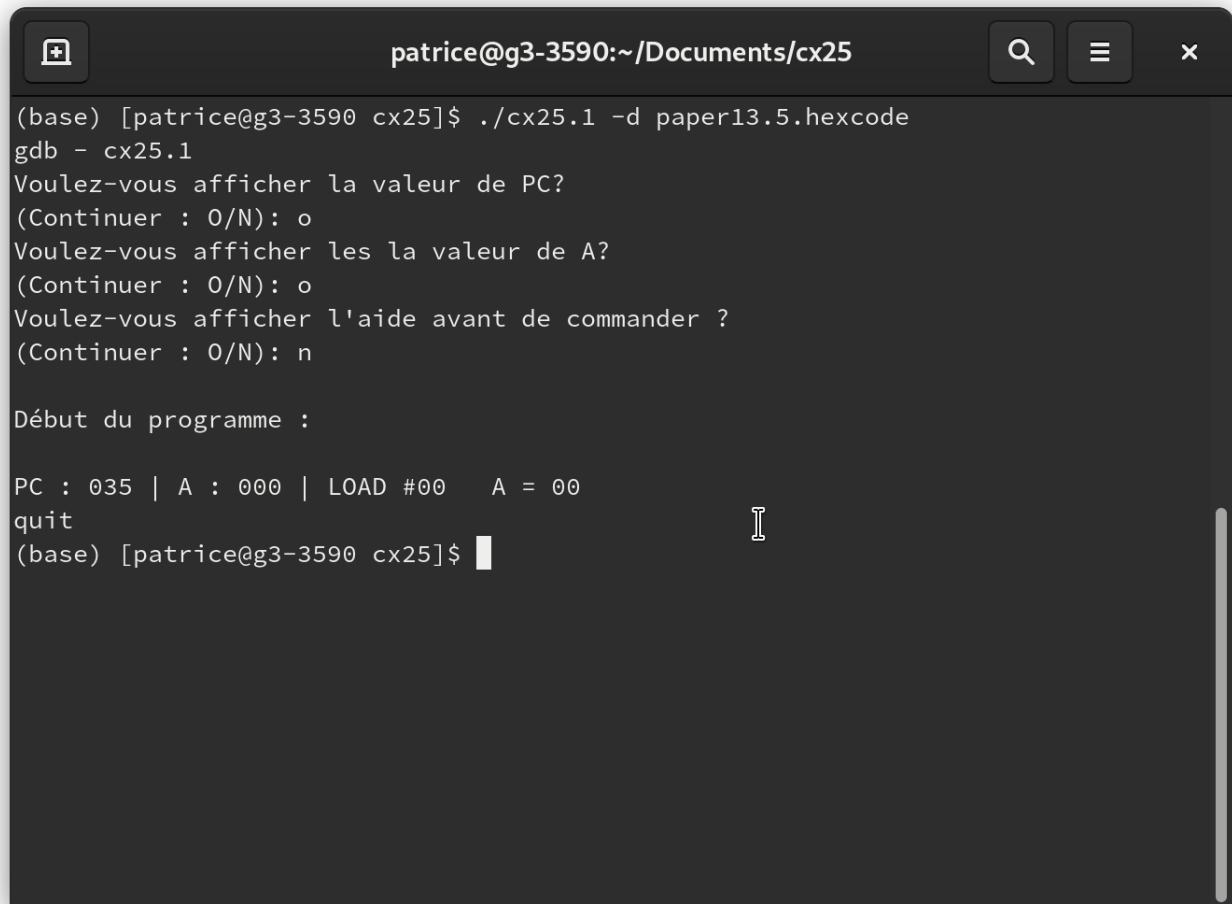
store <adresse> <valeur>;
Signification : enregistre une valeur saisie en hexadecimal à l adresse designe
e. Un fichier denommee : new_version est automatiquement cree avec la modificati
on apportee.
Exemple d utilisation : store 40 2F

quit
Signification : Arrêt du programme.

help
Signification : afficher l'aide.
PC : 037 | A : 000 | STORE 47 data[47] = A
```

FIGURE 6 – Un exemple d'utilisation de la commande help

quit : ferme le programme

A screenshot of a terminal window with a dark background. The window title is 'patrice@g3-3590:~/Documents/cx25'. The terminal shows a sequence of commands and prompts: '(base) [patrice@g3-3590 cx25]\$ ./cx25.1 -d paper13.5.hexcode', 'gdb - cx25.1', 'Voulez-vous afficher la valeur de PC?', '(Continuer : O/N): o', 'Voulez-vous afficher les la valeur de A?', '(Continuer : O/N): o', 'Voulez-vous afficher l'aide avant de commander ?', '(Continuer : O/N): n', 'Début du programme :', 'PC : 035 | A : 000 | LOAD #00 A = 00', 'quit', and finally '(base) [patrice@g3-3590 cx25]\$' followed by a cursor. The window has standard macOS window controls (zoom, search, menu, close) in the top right corner.

```
(base) [patrice@g3-3590 cx25]$ ./cx25.1 -d paper13.5.hexcode
gdb - cx25.1
Voulez-vous afficher la valeur de PC?
(Continuer : O/N): o
Voulez-vous afficher les la valeur de A?
(Continuer : O/N): o
Voulez-vous afficher l'aide avant de commander ?
(Continuer : O/N): n

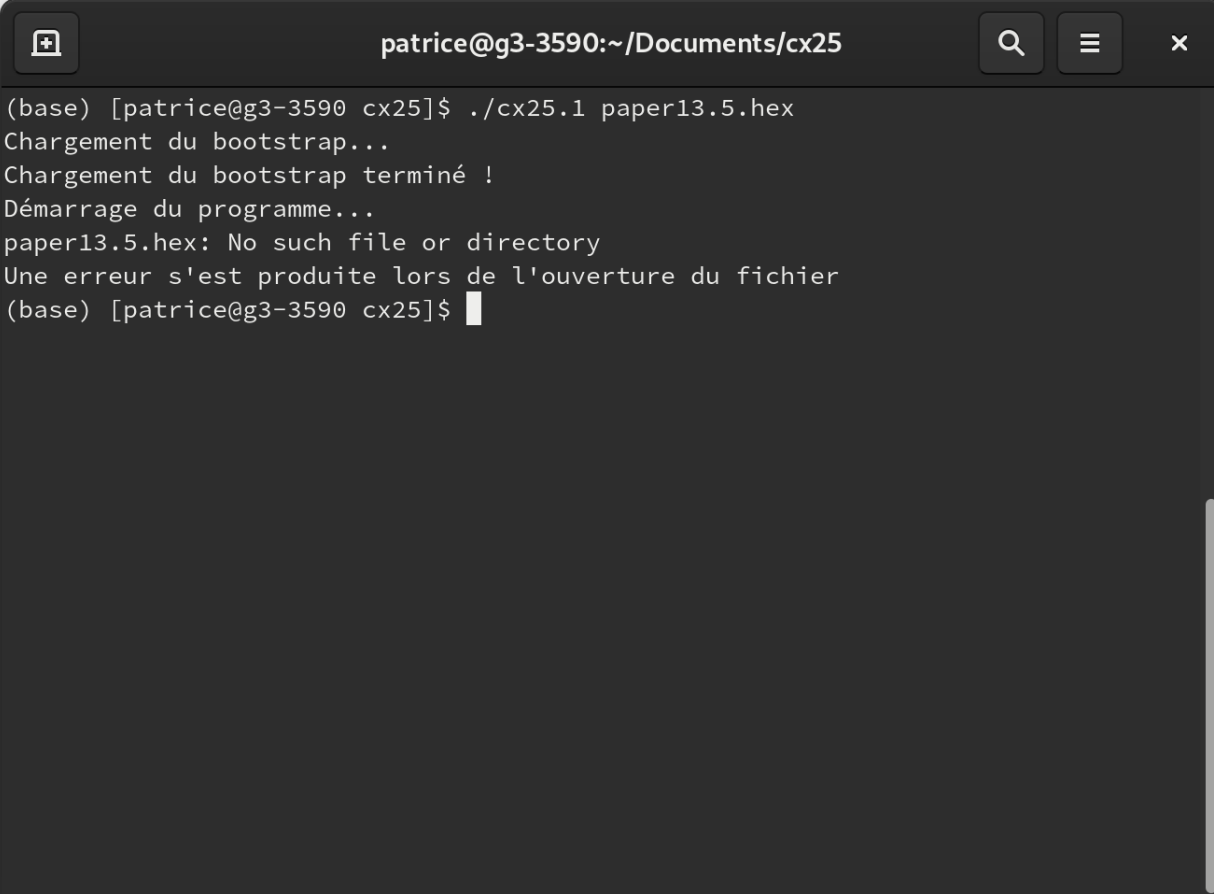
Début du programme :

PC : 035 | A : 000 | LOAD #00 A = 00
quit
(base) [patrice@g3-3590 cx25]$
```

FIGURE 7 – Un exemple d'utilisation de la commande quit

## 4 Quelles erreurs peut-on rencontrer ?

Échec lors de l'ouverture du fichier :

A terminal window with a dark background and light gray text. The window title is 'patrice@g3-3590:~/Documents/cx25'. The terminal shows the following sequence of events: a user enters a command to run a program with a file name; the program prints status messages; it then fails to open the file because of a typo in the filename. The error message is displayed in two lines. The prompt returns to the user.

```
(base) [patrice@g3-3590 cx25]$ ./cx25.1 paper13.5.hex
Chargement du bootstrap...
Chargement du bootstrap terminé !
Démarrage du programme...
paper13.5.hex: No such file or directory
Une erreur s'est produite lors de l'ouverture du fichier
(base) [patrice@g3-3590 cx25]$
```

FIGURE 8 – Une erreur liée à une faute d'orthographe lors du passage du nom de fichier en argument

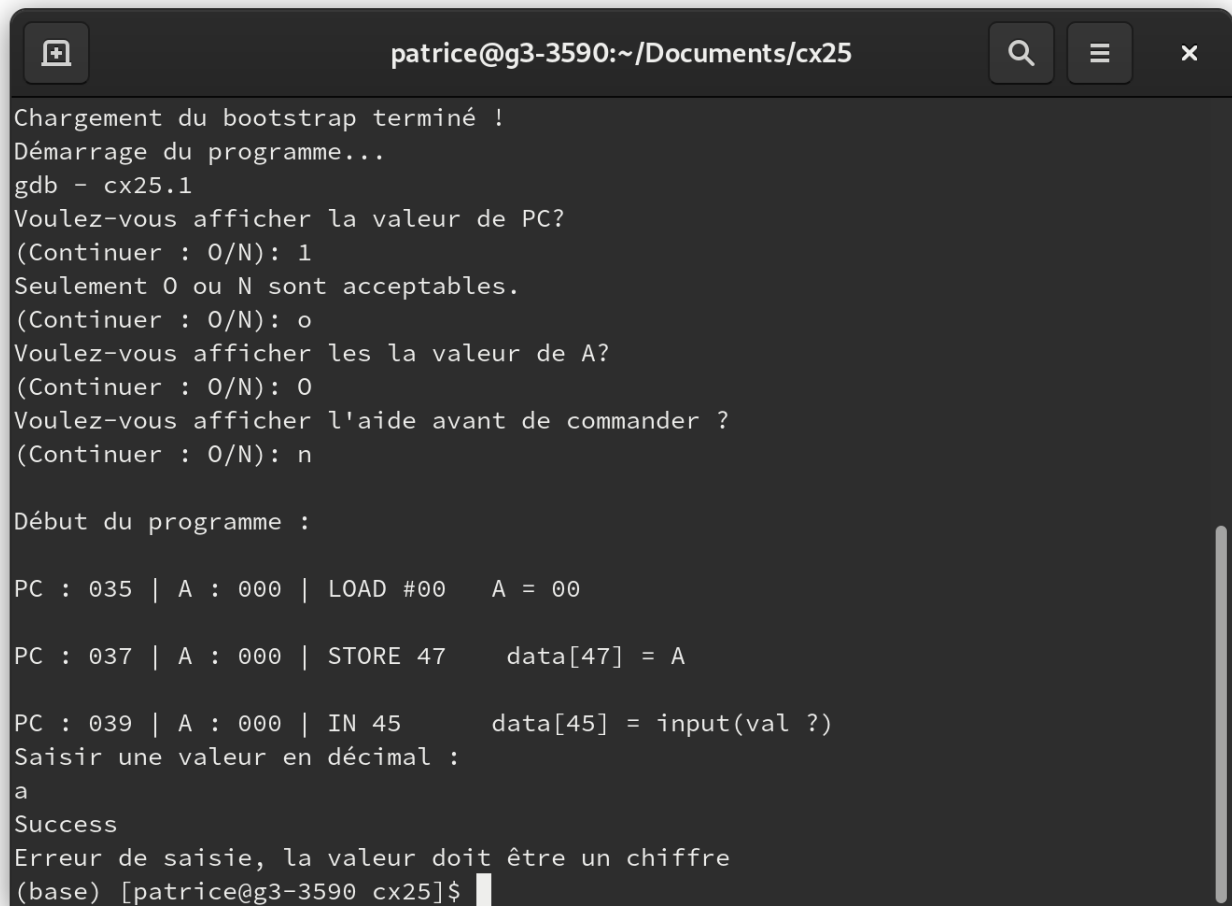
Erreur liée à un mauvais usage des arguments :

A terminal window titled 'patrice@g3-3590:~/Documents/cx25' with search, menu, and close icons in the title bar. The terminal shows the execution of './cx25.1' and './cx25.1 -d paper13.5.hexcode fichier', both of which succeed. The third command './cx25.1 -d' is partially shown, followed by a cursor, indicating an error due to incorrect arguments.

```
(base) [patrice@g3-3590 cx25]$ ./cx25.1
Chargement du bootstrap...
Chargement du bootstrap terminé !
Démarrage du programme...
Success
Usage : <nom du programme> [-d] <nom du fichier>
(base) [patrice@g3-3590 cx25]$ ./cx25.1 -d paper13.5.hexcode fichier
Chargement du bootstrap...
Chargement du bootstrap terminé !
Démarrage du programme...
Success
Usage : <nom du programme> [-d] <nom du fichier>
(base) [patrice@g3-3590 cx25]$ ./cx25.1 -d
Chargement du bootstrap...
Chargement du bootstrap terminé !
Démarrage du programme...
Success
Usage : <nom du programme> [-d] <nom du fichier>
(base) [patrice@g3-3590 cx25]$
```

FIGURE 9 – Exemple d'usage des arguments induisant une erreur

Erreur liée à l'utilisation d'une lettre au lieu d'un chiffre attendu ou l'inverse :



```
patrice@g3-3590:~/Documents/cx25
Chargement du bootstrap terminé !
Démarrage du programme...
gdb - cx25.1
Voulez-vous afficher la valeur de PC?
(Continuer : O/N): 1
Seulement 0 ou N sont acceptables.
(Continuer : O/N): o
Voulez-vous afficher les la valeur de A?
(Continuer : O/N): 0
Voulez-vous afficher l'aide avant de commander ?
(Continuer : O/N): n

Début du programme :

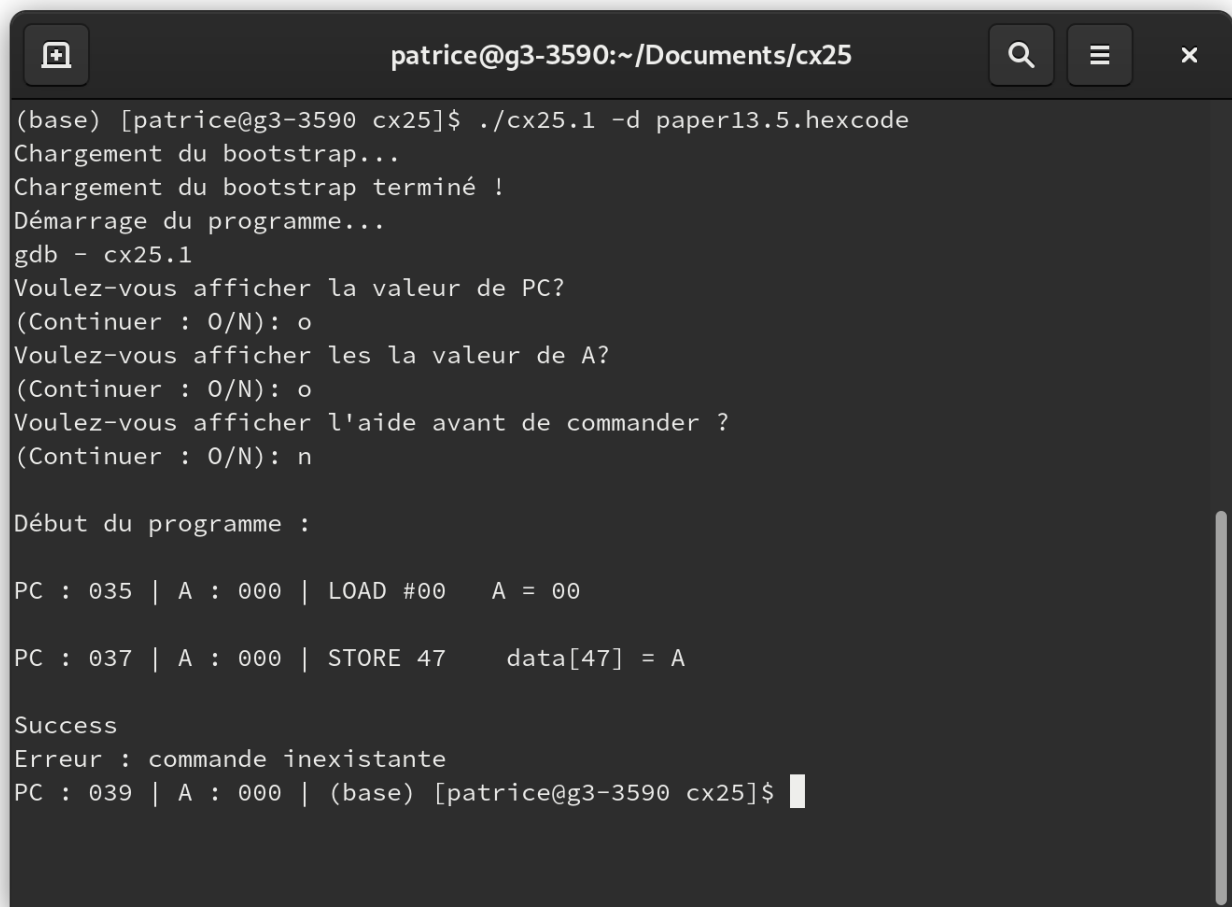
PC : 035 | A : 000 | LOAD #00    A = 00

PC : 037 | A : 000 | STORE 47    data[47] = A

PC : 039 | A : 000 | IN 45       data[45] = input(val ?)
Saisir une valeur en décimal :
a
Success
Erreur de saisie, la valeur doit être un chiffre
(base) [patrice@g3-3590 cx25]$
```

FIGURE 10 – Exemple d'erreur de saisie

Erreur liée à une instruction intégrée au fichier qui ne correspond pas à une commande du langage :



```
patrice@g3-3590:~/Documents/cx25
(base) [patrice@g3-3590 cx25]$ ./cx25.1 -d paper13.5.hexcode
Chargement du bootstrap...
Chargement du bootstrap terminé !
Démarrage du programme...
gdb - cx25.1
Voulez-vous afficher la valeur de PC?
(Continuer : O/N): o
Voulez-vous afficher les la valeur de A?
(Continuer : O/N): o
Voulez-vous afficher l'aide avant de commander ?
(Continuer : O/N): n

Début du programme :

PC : 035 | A : 000 | LOAD #00    A = 00

PC : 037 | A : 000 | STORE 47    data[47] = A

Success
Erreur : commande inexistante
PC : 039 | A : 000 | (base) [patrice@g3-3590 cx25]$
```

FIGURE 11 – Exemple d'erreur dans le fichier paper13.5.hexcode présentant une commande non reconnue par le langage