MINISHELL

[FR] ATTENTION

Les comportements présents dans cet Excel <u>ne sont pas tous a gérer</u> dans la Mandatory du projet Minishell!

Ne vous mettez pas la pression pour essayer de tous les gérer!

Vérifiez juste pour ces cas non gerables s'ils ne Segfault pas, ou n'effectuent pas un comportement suspect!

Utilisez votre bon sens pour exploiter au mieux ce document et faire un Minishell pas trop ambitieux, selon votre situation du projet.

(Par contre, si vous etes chauds pour tout gérer, GG à vous !)

Bon courage et n'hésitez pas à m'envoyer un MP si besoin!

[EN] WARNING

The behaviors listed in this Excel file <u>are not all required to be handled</u> in the mandatory part of the Minishell project!

Don't stress yourself out trying to handle all of them!

Just make sure that these unmanageable cases don't cause a segfault or exhibit suspicious behavior!

Use your common sense to make the best use of this document and create a Minishell that is not overly ambitious, depending on your project situation.

(However, if you're up for handling everything, GG to you!)

Good luck, and feel free to DM me if needed!

(GitHub: vietdu91 / Discord: vietdu91#6048 / Intra: emtran)

Gestion des comportements

	A faire	A tester	Leaks?	Fonctionne		G III / 1 43\	
test.c	Х	X	ок ко	ок ко	Comportement attendu	Sortie d'exit (echo \$?)	Notes
\$> \n (touche entrée)			ОК	ОК	\$>	Exit de l'ancienne commande	
\$> [que des espaces]			ОК	ОК		0	
<pre>\$> [que des tabulations]</pre>			ОК	ОК		0	Crl + V + Tab
\$>:			ОК	ОК		0	Marche aussi pour "#"
\$>!			ОК	ОК		1	
\$> >			ОК	ОК	bash: syntax error near unexpected token `newline'	2	
\$> <			ОК	ОК	\$>	2	
\$> >>			ОК	ОК		2	
\$> <<			ОК	ОК		2	
\$> <>			ОК	ОК		2	
\$> >>>>			ОК	ОК	<pre>bash: syntax error near unexpected token `>>' \$></pre>	2	
\$> >>>>>>>>			ОК		<pre>bash: syntax error near unexpected token `>>' \$></pre>	2	
\$> <<<<			ОК	ОК	<pre>bash: syntax error near unexpected token `<<' \$></pre>	2	

	\$> <<<<<<<<		0	ОК	bash: syntax error near unexpected token `<<' \$>	2	
	\$> > > >		0	ОК	bash: syntax error near unexpected token `>' \$>	2	Marche aussi pour <
	\$> >> >> >>		0	ОК	bash: syntax error near unexpected token `>>' \$>	2	Marche aussi pour <
	\$> >>> >>		0	ОК	bash: syntax error near unexpected token `>>' \$>	2	Marche aussi pour <
	\$> /		0	ОК	bash: /: Is a directory \$>	126	
	\$> //		0	ОК	bash: //: Is a directory \$>	126	
	\$> /.		0	ОК	bash: /.: Is a directory S>	126	
	\$> /.///		0	ОК	bash: /.///: Is a directory \$>	126	
	\$>		0	ОК	bash: /////: Is a directory	126	
	\$> \\	NON GERE	0	ОК	bash: \: command not found	127	
	\$> \\\\	NON GERE	0	ОК	bash: \\: command not found	127	
	\$> \\\\\\	NON GERE	0	ОК	bash: \\\\: command not found	127	
	\$> -		0	ОК	bash: -: command not found \$>	127	Marche aussi pour "_@°]=+£\$?"
	\$>		0	ОК	bash: syntax error near unexpected token ` ' \$>	2	Marche aussi pour les metacharacters "&();"
	\$> hola		0	ОК	bash: syntax error near unexpected token ` ' \$>	2	
rAXE)	\$>		O	ОК	<pre>bash: syntax error near unexpected token ` ' \$></pre>	2	
LEE (SYN'	\$> [[0	ОК	<pre>bash: syntax error near unexpected token ` ' \$></pre>	2	
A LA VOI	\$>		O	ОК	<pre>bash: syntax error near unexpected token ` ' \$></pre>	2	
CARACTERES	\$>		0	ОК	<pre>bash: syntax error near unexpected token ` ' \$></pre>	2	
CAR	\$> >> ><		0	ОК	bash: syntax error near unexpected token ` ' \$>	2	
	\$> &&		0	ОК	bash: syntax error near unexpected token `&&' \$>	2	
	\$> &&&&&		0	ОК	bash: syntax error near unexpected token `&&' \$>	2	
	\$> &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&		0	ОК	bash: syntax error near unexpected token `&&' \$>	2	
	\$> ;;	NON GERE	0	ОК	bash: syntax error near unexpected token `;;' \$>	2	
	\$> ;;;;;	NON GERE	0	ОК	<pre>bash: syntax error near unexpected token `;;' \$> bash: syntax error near</pre>	2	
	\$> ;;;;;;;;;; \$> ()	NON GERE	0	ОК	unexpected token `;;' \$> bash: syntax error near	2	
		NON GERE	0	ОК	unexpected token `)' \$>	2	
	\$> (())	NON GERE	0	ОК	<pre>bash: syntax error near unexpected token `)' \$></pre>	2	

\$> (((())))	NON GERE	C	ОК		<pre>bash: syntax error near unexpected token `)' \$></pre>	2	
\$> "	П	NON GERE	C	ОК	OK	bash: : command not found \$>	127	
\$> "	hola"		C	ОК		bash: hola: command not found \$>	127	
\$> '	hola'		C	ОК		bash: hola: command not found \$>	127	
\$> '		NON GERE	C	ОК		bash: : command not found \$>	127	
\$> *		NON GERE	C	ОК		bash: crashtest.c: command not found \$>	127	[BONUS]
\$> *	/*	NON GERE	C	ОК	OK	bash: Docs/attest.cation: command not found \$>	127	[BONUS]
\$> *	/*	NON GERE	C	Ж		bash: */*: No such file or directory	127	[BONUS] S'il n'y aucun dossier à l'intérieur
\$> .		NON GERE	C	ОК	ОК	bash: .: filename argument required \$>	2	
\$> .		NON GERE	C	ОК		bash:: command not found \$>	127	
\$> ~		NON GERE	C	ОК		bash: /home/vietdu91: Is a directory \$>	126	
\$> A	BC=hola	NON GERE	C	ОК		\$>	0	/!\ A gerer comme une erreur
\$> 4	ABC=hola	NON GERE	C	ОК	ОК	bash: 4ABC=hola: command not found \$>	127	
hola			C	ОК		bash: hola: command not found \$>	127	
\$> h	ola que tal		C	ОК	OK	sash: hola: command not found \$>	127	
\$> M	akefile		C	ОК	OK	bash: Makefile: command not found \$>	127	
\$> e	cho		C	ОК	OK	\$>	0	
\$> e	cho -n		C	OK .		\$>	0	
\$> e	cho Hola		C	ОК		Hola \$>	0	
\$> e	choHola		C	ОК	OK	bash: echoHola: command not found \$>	127	
\$> e	cho-nHola		C	ОК	OK	bash: echo-nHola: command not found \$>	127	
\$> e	cho -n Hola		C	OK .		Hola\$>	0	
\$> e	cho "-n" Hola		C	ОК	ОК	Hola\$>	0	
\$> e	cho -nHola		C	СК		-nHola \$>	0	
\$> e	cho Hola -n		C	Ж	ОК	Hola -n \$>	0	
\$> e	cho Hola Que Tal		C	ОК	OK	Hola Que Tal \$>	0	
\$> e	cho Hola		C	ОК		Hola \$>	0	
\$> e	cho Hola Que Tal		C	ОК	OK	Hola Que Tal \$>	0	
\$> e	cho \n hola		C	ОК	OK	n hola \$>	0	A gerer ?
\$> e	cho " " cat -e		C	ОК	OK	\$ \$>	0	
\$> e	cho cat -e		C	ОК	OK	\$ \$>	0	
\$> "	"''echo hola""''' que""'' tal""''		C	ОК	OK	hola que tal \$>	0	
\$> e	cho -n -n		C	OK .		\$>	0	
\$> e	cho -n -n Hola Que		C	Ж	ОК	Hola Que\$>	0	
\$> e	cho -p		C	ОК	ОК	-p \$>	0	
\$> e	cho -nnnnn		C	ОК		\$>	0	
\$> e	cho -n -nnn -nnnn		C	СК	ОК	\$>	0	

\$> echo -n-nnn -nnnn	ОК	ок	-n-nnn -nnnn \$>	0	
\$> echo -n -nnn hola -nnnn	ОК	ОК	hola -nnnn\$>	0	
\$> echo -n -nnn-nnnn	ОК	ОК	-nnn-nnnn\$>	0	
\$> echon	ОК	ОК	n \$>	0	
\$> echo -nnnn	ОК	ОК	n\$>	0	
\$> echo -nnnnnnn	ОК	ОК	nnnnnn\$>	0	
\$> echo -nnnnnnn	ОК	ОК	nnnn\$>	0	
\$> echo \$	ОК	ОК	\$ \$>	<u> </u>	
\$> echo \$?	ОК	ок	0 \$>	0	Renvoie le code de retour du dernier pipeline exécuté à l'avant-plan
\$> echo \$?\$	ОК	ОК	0\$ \$>	0	
\$> echo \$? echo \$? echo \$?	ОК	ОК	0 \$>	0	
\$> echo \$:\$= cat -e	ОК	ок	\$:\$=\$ \$>	0	
\$> echo " \$ " cat -e	ОК	ОК	\$ \$ \$>	0	
\$> echo ' \$ ' cat -e	ок	ок	\$ \$ \$>	0	
\$> echo \$HOME	ок	ок	/home/vietdu91 \$>	0	
\$> echo \\$HOME	ок	ОК	\$HOME \$>	0	
<pre>\$> echo my shit terminal is [\$TERM]</pre>	ОК	ОК	my shit terminal is [xterm-256color] \$>	0	
<pre>\$> echo my shit terminal is [\$TERM4</pre>	ОК	ок	my shit terminal is [\$>	0	
<pre>\$> echo my shit terminal is [\$TERM4]</pre>	ок	ок	my shit terminal is [] \$>	0	
\$> echo \$UID	ок	ОК	1000 \$>	0	
\$> echo \$HOME9	ОК	ок	\$>	0	Marche pour tous les caractères alphanumériques de l'ASCII
\$> echo \$9H0ME	ОК	ок	HOME \$>	0	
\$> echo \$HOME%	ок	ОК	/home/vietdu91% \$>	0	Marche pour les autres caractères (ponctuation, caractères non-ASCII, etc.)
\$> echo \$UID\$HOME	ОК	ок	1000/home/vietdu91% \$>	0	
<pre>\$> echo Le path de mon HOME est \$HOME</pre>	ОК	ОК	Le path de mon HOME est /home/vietdu91 \$>	0	
<pre>\$> echo \$USER\$var\\$USER\$USER\\$USER</pre>	ОК	ОК	vietdu91\$USERvietdu91\$USERtestvietdu91 \$>	0	A ne pas gerer si '\' non gere : Doit sortir : vietdu91vietdu91vietdu91
<pre>\$> echo \$hola*</pre>	ок	ОК	\$>	0	Cas d'un variable inexistant
\$> echo -nnnn \$hola	ОК	ОК	\$>	0	
\$> echo > <	ОК	ок	<pre>bash: syntax error near unexpected token `<' \$></pre>	2	
\$> echo	ок	ОК	<pre>bash: syntax error near unexpected token ` ' \$></pre>	2	
\$> EechoE	ок	ок	<pre>bash: EechoE: command not found \$></pre>	127	
\$> .echo.	ок	ок	<pre>bash: .echo.: command not found \$></pre>	127	
\$> >echo>	ОК	ок	<pre>bash: syntax error near unexpected token `newline' \$></pre>	2	
\$> <echo<< td=""><td>ОК</td><td>ОК</td><td><pre>bash: syntax error near unexpected token `newline' \$></pre></td><td>2</td><td></td></echo<<>	ОК	ОК	<pre>bash: syntax error near unexpected token `newline' \$></pre>	2	
\$> >>echo>>	ОК	ОК	bash: syntax error near unexpected token `newline' \$>	2	
\$> echo	ОК	ОК	<pre>bash: syntax error near unexpected token ` ' \$></pre>	2	
\$> echo -n hola	ОК	ок	<pre>bash: syntax error near unexpected token ` ' \$></pre>	2	

	\$> echo *	NON GERE	ОК	ОК	crashtest.c Docs Dockers Drawings Makefile minishell 0 philosophers srcs test.c \$>	[BONUS]
ЕСНО	\$> echo '*'	NON GERE	ОК	ОК	* 0 \$>	Marche aussi pour les double quotes
Ш	\$> echo D*	NON GERE	ОК	ОК	Docs Dockers Drawings 0 \$>	[BONUS]
	\$> echo *Z	NON GERE	ОК	ОК	*Z \$>	[BONUS]
	\$> echo *t hola	NON GERE	ОК	ок	<pre>crashtest.c test.c hola \$></pre>	[BONUS]
	\$> echo *t	NON GERE	ок		<pre>crashtest.c test.c \$></pre>	[BONUS]
	\$> echo \$*	NON GERE	ОК	ок	\$>	[BONUS]
	<pre>\$> echo hola*hola *</pre>	NON GERE	ок	ОК	hola*hola crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$>	[BONUS]
	\$> echo \$hola*	NON GERE	ок	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell 0 philosophers srcs test.c \$></pre>	[BONUS]
	\$> echo \$HOME*	NON GERE	ок	ок	/home/vietdu91 0 \$>	[BONUS]
	\$> echo \$""		ок	ОК	\$>	
	\$> echo "\$"""		ОК	UK	\$ \$>	
	\$> echo '\$'''		ОК		\$ \$>	
	\$> echo \$"HOME"		ОК		HOME \$>	
	\$> echo \$''HOME		ОК		HOME \$>	
	\$> echo \$""HOME		ОК	ОК	HOME \$>	
	\$> echo "\$H0"ME		ОК	ОК	ME \$>	
	\$> echo '\$H0'ME		ОК		\$HOME	
	\$> echo "\$H0""ME"		ОК		ME	
	\$> echo '\$H0''ME'		ОК	ОК	\$HOME	
	\$> echo "'\$HO''ME'"		ОК		''ME' \$>	Il y a bien 3 singles quotes avant le ME. Je ne sais pas pourquoi Excel tronque à l'affichage le 3e quote
	\$> echo ""\$HOME		ОК	ОК	/home/vietdu91 \$>	
	\$> echo "" \$HOME		ОК	ОК	/home/vietdu91 \$>	
	\$> echo ''\$HOME		ОК	ОК	/home/vietdu91 \$>	
	\$> echo '' \$HOME		ОК		<pre>/home/vietdu91 \$></pre>	
	\$> echo \$"HO""ME"		ОК		HOME \$>	
	\$> echo \$'HO''ME'		ОК	OK .	HOME \$>	
	\$> echo \$'HOME'		ОК		HOME \$>	
	\$> echo "\$"HOME		ОК	OK .	\$HOME	
	\$> echo \$=HOME		ОК	OK .	\$=HOME \$>	
	\$> echo \$"HOLA"		ОК		HOLA \$>	
	\$> echo \$'HOLA'		ОК	OK .	HOLA \$>	
	\$> echo \$DONTEXIST Hola		ОК		Hola \$>	Le cas ou la variable n'existe pas et n'est pas importee
	\$> echo "hola"		ОК	ОК	hola \$>	

\$> echo 'hola'	ОК	ОК	hola \$>	0	
\$> echo ''hola''	ОК	OV	hola \$>	0	
\$> echo ''h'o'la''	ОК	OK	hola \$>	0	
\$> echo "''h'o'la''"	ОК	OV	'h'o'la'' \$>	0	
\$> echo "'"h'o'la"'"	ОК	OK	hola' \$>	0	
\$> echo"'hola'"	ОК	OK	bash: echo'hola': command not found \$>	0	
\$> echo "'hola'"	ОК	OK	'hola' \$>	0	
\$> echo '"hola"'	ОК	OK	"hola" \$>	0	
\$> echo '''ho"'''l"a'''	ОК	01/	ho"l"a \$>	0	
\$> echo hola""""""""""""""""""""""""""""""""""""	ОК	ок	hola \$>	0	
\$> echo hola"'''"	ОК	OK	hola'''''' \$>	0	
\$> echo hola'''''	ОК	OV	hola \$>	0	
\$> echo hola'""""""""	ОК		hola""""""" \$>	0	
\$> e"cho hola"	ОК		bash: echo hola: command not found \$>	127	
\$> e'cho hola'	ОК		bash: echo hola: command not found \$>	127	
\$> echo "hola " cat -e	ОК	ОК	hola \$ \$>	0	
\$> echo ""hola	ОК	ОК	hola \$>	0	
\$> echo "" hola	ОК	ОК	hola \$>	0	
\$> echo "" hola	ОК	ок	hola \$>	0	
\$> echo ""hola	ок	ок	hola \$>	0	
\$> echo "" hola	ок	ок	hola \$>	0	
\$> echo hola""bonjour	ОК	ок	holabonjour \$>	0	
\$> "e"'c'ho 'b'"o"nj"o"'u'r	ОК	ОК	bonjour \$>	0	
\$> ""e"'c'ho 'b'"o"nj"o"'u'r"	ОК		bash: e'c'ho 'b'onjo'u'r: command not found \$>	127	
\$> echo "\$DONTEXIST"Makefile	ок	ок	Makefile \$>	0	
<pre>\$> echo "\$DONTEXIST""Makefile"</pre>	ок	ок	Makefile \$>	0	
<pre>\$> echo "\$DONTEXIST" "Makefile"</pre>	ОК	ОК	Makefile \$>	0	Attention a l'espace avant
\$> \$?	ОК	OK	bash: 0: command not found \$>	127	Renvoie le code de retour du dernier pipeline exécuté à l'avant-plan
\$> \$?\$?	ОК	OK	bash: 00: command not found \$>	127	
\$> ?\$HOME	ОК	O.K	<pre>bash: ?/home/vietdu91: command not found \$></pre>	127	
\$> \$	ОК	OK .	<pre>bash: \$: command not found \$></pre>	127	
\$> \$HOME	ОК	ОК	<pre>bash: /home/vietdu91: Is a directory \$></pre>	126	
\$> \$HOMEdskjhfkdshfsd	ОК		\$>	0	
\$> "\$HOMEdskjhfkdshfsd"	ОК	OK	bash: : command not found \$>	127	
\$> '\$HOMEdskjhfkdshfsd'	OK	OK .	<pre>bash: \$HOMEsddfddfssfdfs: command not found \$></pre>	127	Con diverse della di la di
\$> \$DONTEXIST	ОК	OK	\$>	0	Cas d'un variable inexistant

	\$> \$LESS\$VAR		ОК	UK	bash: -R: command not found \$>	127	
	\$> Ctlr-C		ОК	OK	\$> ^C \$>	130	
	<pre>\$> holaCtlr-C</pre>		ОК	OK	\$> hola^C \$>	130	
	<pre>\$> cat (faire Ctlr-C apres avoir fait plusieurs fois [ENTREE])</pre>				<u>*</u>	130	
			ОК	ОК	^c		
	<pre>\$> cat rev(faire Ctlr-C apres avoir fait plusieurs fois [ENTREE])</pre>				\$>	0	
×	plusieurs fois [ENTREE])		ОК	ОК	^c		
GNAUX	\$> Ctlr-D				\$> \$> exit	0	Cost du occarama
<u>5</u>			ОК		~/minishell \$>		Sort du programme
S	<pre>\$> hola Ctlr-D \$> Ctlr-\</pre>		OK	OK	\$> hola		Ne fait rien
	\$> hola Ctlr-\		ок ок	ок ок	\$> hola	<i>1</i>	Ne latt i tell
	\$> cat (faire Ctlr-\ apres avoir fait plusieurs		UK	OK .	- 110 to	131	
	fois [ENTREE])		ОК	ОК			
					^\Quit (core dumped) \$>		
	\$> Ctlr-Z		ОК	ОК	\$>	1	Ne fait rien
	<pre>\$> sleep 3 sleep 3 sleep 3 (faire Ctlr-C une seconde apres)</pre>		ОК	ОК	^C \$>	130	
	\$> sleep 3 sleep 3 sleep 3 (faire Ctlr-D une seconde apres)		ОК		\$>	0	Ne fait rien
	\$> sleep 3 sleep 3 sleep 3 (faire Ctlr-\ une seconde apres)		ОК	ОК	^\Quit (core dumped) \$>	131	
	\$> env		ок	ОК	PATH= \$>	0	
	\$> env hola	NON GERE	ок		env: 'hola': No such file or directory \$>	127	
	\$> env hola que tal	NON GERE	ОК	ОК	env: 'hola': No such file or directory \$>	127	
	\$> env env	NON GERE	ок	ОК	PATH= \$>	0	Affiche une seule fois env
	\$> env env env env	NON GERE	ОК	ОК	PATH= \$>	0	Affiche une seule fois env
	\$> env ls	NON GERE	ок	ок	crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$>	0	
	\$> env ./Makefile	NON GERE	ОК	OK	env: './Makefile': Permission denied \$>	126	
	<pre>\$> export HOLA=bonjour \$> env</pre>		ОК	ОК	HOLA=bonjour \$>	0 0	
	<pre>\$> export HOLA=bonjour \$> env</pre>		ок	ОК	 HOLA=bonjour \$>	0 0	
	\$> export		ок		export PATH="" \$>		Les variables sont triees par ordre alphabetique On n'oublie pas les doubles quotes qui designent les strings attribues a chaque variable
	<pre>\$> export Hola \$> export</pre>		ОК	OK	<pre> export HOME="/home/vietdu91" export Hola export PATH="" \$></pre>	0	Hola n'existe pas dans env mais il existe dans export
	<pre>\$> export Hola9hey \$> export</pre>		ОК	ОК	export HOME="/home/vietdu91" export Hola9hey export PATH="" \$>	0 0	

<pre>\$> export \$DONTEXIST</pre>		ОК	ОК	export HOME="/home/vietdu91" export PATH="" \$>		Le cas ou la variable n'existe pas et n'est pas importee
\$> export grep "HOME"		ОК	ОК	export HOME="/home/vietdu91" \$>	Ð	
\$> export ""		ОК	ОК	bash: export: `': not a valid identifier 1	1	
\$> export =		ОК	ОК	bash: export: '=': not a valid identifier 1	1	
\$> export %		ОК	ОК	bash: export: '%': not a valid identifier	1	
\$> export \$?		OK	ОК	\$> bash: export: `0': not a valid identifier 1	1	
\$> export ?=2		OK	ОК	bash: export: '?=2': not a valid identifier 1	1	
\$> export 9HOLA=		OK	ОК	bash: export: '9HOLA=': not a valid identifier 1	1	
<pre>\$> export HOLA9=bonjour \$> env</pre>		ок	ОК	\$> 6 HOLA9=bonjour 5>	9	
<pre>\$> export _HOLA=bonjour \$> env</pre>		ок	ОК	0 HOLA=bonjour 5>	0	
<pre>\$> exportHOLA=bonjour \$> env</pre>		ок	ОК	HOLA=bonjour \$>	9	
<pre>\$> export _HO_LA_=bonjour \$> env</pre>		ок	ОК	 _HO_LA_=bonjour \$>	0	
<pre>\$> export HOL@=bonjour</pre>		ок	ок	bash: export: 'HOL@=bonjour': not a valid identifier \$>	1	Cas où le nom de variable ne contient pas de lettres, ni de chiffres, ni d'underscores
\$> export HOL\~A=bonjour NON GERE		ок	ОК	bash: export: `HOL~A=bonjour': not a valid identifier \$>	1	
<pre>\$> export -HOLA=bonjour</pre>		ок			2	
\$> exportHOLA=bonjour		ок	ОК		2	
\$> export HOLA-=bonjour		ок	ОК	<pre>bash: export: `HOLA-=bonjour': not a valid identifier \$></pre>	1	
\$> export HO-LA=bonjour		ок	ОК	bash: export: `HO-LA=bonjour': not a valid identifier \$>	1	
\$> export HOL.A=bonjour		ок	ОК	<pre>bash: export: `HOL.A=bonjour': not a valid identifier \$></pre>	1	
\$> export HOL\\\\$A=bonjour NON GERE		ок	ОК	<pre>bash: export: `HOL\\$A=bonjour': not a valid identifier \$></pre>	1	
\$> export HO\\\\LA=bonjour NON GERE		ок	ОК	<pre>bash: export: `HO\\LA=bonjour': not a valid identifier \$></pre>	1	
<pre>\$> export HOL}A=bonjour</pre>		ок	ОК	<pre>bash: export: `HOL}A=bonjour': not a valid identifier \$></pre>	1	
<pre>\$> export HOL{A=bonjour</pre>		ок	ОК	<pre>bash: export: `HOL{A=bonjour': not a valid identifier \$></pre>	1	
\$> export H0*LA=bonjour		ок	ок	<pre>bash: export: `HO*LA=bonjour': not a valid identifier \$></pre>	1	
\$> export HO#LA=bonjour		ок	ок	bash: export: `HO#LA=bonjour': not a valid identifier \$>	1	
\$> export HO@LA=bonjour		ок	ок	bash: export: `HO@LA=bonjour': not a valid identifier \$>	1	
\$> export HO!LA=bonjour		ок	ОК	bash: !LA=bonjour: event not found	9	
<pre>\$> export HO\$?LA=bonjour \$> env</pre>		ок	ОК	HOOLA=bonjour	9	
\$> export +HOLA=bonjour		ОК	ОК	bash: export: `+HOLA=bonjour': not a valid identifier \$>	ı	
\$> export HOL+A=bonjour		ок	ОК	bash: export: `HOL+A=bonjour': not a valid identifier \$>	1	
<pre>\$> export HOLA+=bonjour \$> env</pre> NON GERE		ок	ОК	HOLA=bonjour \$>	9	
<pre>\$> export HOLA=bonjour \$> export HOLA+=bonjour \$> env</pre> NON GERE		ок	ОК	HOLA=bonjourbonjour \$>)	"+=" s'ajoute au contenu de la variable

% UNSET 2 EXPO N N

	<pre>\$> exportHOLA=bonjour</pre>						0	
	\$> exporthola=bonjour \$> env	NON GERE		ОК	ОК	PATH= \$>	Θ	
	<pre>\$> export HOLA =bonjour</pre>			ОК	ок	bash: export: '=bonjour': not a valid identifier \$>	1	
	<pre>\$> export HOLA = bonjour</pre>			ОК	ОК	<pre>bash: export: '=': not a valid identifier \$></pre>	1	
	<pre>\$> export HOLA=bon jour \$> env</pre>			ОК	ОК	 HOLA=bon \$>	0	
	<pre>\$> export HOLA= bonjour \$> env</pre>			ОК	ОК	 HOLA= \$>	0	
	<pre>\$> export HOLA=bonsoir \$> export HOLA=bonretour \$> export HOLA=bonjour \$> env</pre>			ОК	ОК	 HOLA=bonjour \$>	0 0 0	
	<pre>\$> export HOLA=\$HOME \$> env</pre>			ОК	ОК	HOLA=/home/vietdu91 \$>	0 0	
	<pre>\$> export HOLA=bonjour\$HOME \$> env</pre>			ОК	ОК	 HOLA=bonjour/home/vietdu91 \$>	0	
	<pre>\$> export HOLA=\$HOMEbonjour \$> env</pre>			ОК	ОК	 HOLA= \$>	0 0	
	<pre>\$> export HOLA=bon\$jour \$> env</pre>			ОК	ОК	 HOLA=bon \$>	0	Le cas où la variable "\$jour" n'existe pas
SET	<pre>\$> export HOLA=bon\jour \$> env</pre>	NON GERE		ОК	ОК	 HOLA=bonjour \$>	0 0	
& UNS	<pre>\$> export HOLA=bon\\jour \$> env</pre>	NON GERE		ОК	ОК	 HOLA=bon\jour \$>	0 0	
RT	<pre>\$> export HOLA=bon(jour</pre>	NON GERE		ОК	ОК	<pre>bash: syntax error near unexpected token `(' \$></pre>	2	Marche pour les caractères : ()
EXPO	<pre>\$> export HOLA=bon()jour</pre>	NON GERE		ОК	ОК	bash: syntax error near unexpected token `(' \$>	2	
ENV &	<pre>\$> export HOLA=bon&jour</pre>	NON GERE		ОК	ОК	bash: jour: command not found \$>	127	Marche pour les caractères : &
E	<pre>\$> export HOLA=bon@jour \$> env</pre>			ОК	ок	\$> env HOLA=bon@jour \$>	0	
	<pre>\$> export HOLA=bon;jour \$> env</pre>			ОК	ОК	bash: jour: command not found \$> env HOLA=bon \$>	127 0	
	<pre>\$> export HOLA=bon!jour</pre>			ОК	ОК	bash: !jour: event not found \$>	1	
	<pre>\$> export HOLA=bon"jour" \$> env</pre>			ок	ОК	 HOLA=bonjour PATH= \$>	0 0	
	<pre>\$> export HOLA\$USER=bonjour \$> env</pre>			ОК	ок	HOLAvietdu91=bonjour PATH= \$>	0	
	<pre>\$> export HOLA=bonjour=casse-toi \$> echo \$HOLA</pre>			ОК	ОК	HOLA=bonjour=casse-toi \$>	0 0	
	\$> export "HOLA=bonjour"=casse-toi \$> echo \$HOLA			ОК	ОК	HOLA=bonjour=casse-toi \$>	0	
	<pre>\$> export HOLA=bonjour \$> export BYE=casse-toi \$> echo \$HOLA et \$BYE</pre>			ОК	ОК	bonjour et casse-toi \$>	0 0 0	
	<pre>\$> export HOLA=bonjour BYE=casse-toi \$> echo \$HOLA et \$BYE</pre>			ОК	ОК	bonjour et casse-toi \$>	0 0	
	<pre>\$> export \$HOLA=bonjour \$> env</pre>			ОК	OK	bonjour=bonjour PATH= \$>	0	

<pre>\$> export HOLA="bonjour" \$> echo \$HOLA cat -e</pre>		ОК	ОК	bonjour\$ \$>	9	
\$> export HOLA=" -n bonjour " \$> echo \$HOLA		ОК	ОК		9	
\$> export HOLA="bonjour "/ \$> echo \$HOLA		ок	ОК	bonjour / \$>	9	
\$> export HOLA='"' \$> echo " \$HOLA " cat -e		ок	ОК	" \$ \$>	9	
\$> export HOLA=at \$> c\$HOLA Makefile		ок	ОК	[cela affiche le Makefile] \$>	9	
\$> export "" HOLA=bonjour \$> env				bash: export: `": not a valid identifier \$> env	1	
		OK	OK	 HOLA=bonjour		
\$> export HOLA="cat Makefile grep NAME"		OK	ОК	\$> cat Makefile grep NAME	9	
\$> echo \$H0LA \$> export H0LA=hey \$> echo \$H0LA\$H0LA\$H0LA=hey\$H0LA		OK	ОК	heyheyhey=heyhey	9	
\$> echo \$HOLA\$HOLA=Ney\$HOLA\$ \$> export HOLA=" bonjour hey " \$> echo \$HOLA cat -e		ОК		bonjour hey\$ \$>	9	
\$> echo \$HOLA Cat -e \$> export HOLA=" bonjour hey " \$> echo """\$HOLA""" cat -e		ОК	ОК	bonjour hey \$	9	
\$> echo """\$HULA"" cat -e \$> export HOLA=" bonjour hey " \$> echo wesh"\$HOLA" cat -e		ОК		<pre>\$> wesh bonjour hey \$ \$></pre>	9	
\$> echo wesh \$HOLA Call-e \$> export HOLA=" bonjour hey " \$> echo wesh""\$HOLA.		ОК	ОК	wesh bonjour hey .	9	/!\ Bien supprimer les espaces en trop au debut et a la fin du string d'export
\$> export HOLA=" bonjour hey " \$> echo wesh\$""HOLA.		ОК	ОК	weshHOLA.	9	et a ta full ou struig a export
\$> export HOLA=" bonjour hey " \$> echo wesh\$"HOLA HOLA".		ОК	ОК	weshHOLA HOLA.	9	
\$> export HOLA=bonjour \$> export HOLA=" hola et \$HOLA"		ОК	ОК	hola et bonjour \$>	9	
\$> echo \$HOLA \$> export HOLA=bonjour				hola et \$HOLA	9	
\$> export HOLA=' hola et \$HOLA' \$> echo \$HOLA		ОК	ОК	\$>	5 5 9	
<pre>\$> export H0LA=bonjour \$> export H0LA=" hola et \$H0LA"\$H0LA \$> echo \$H0LA</pre>		ОК	ОК	hola et bonjourbonjour \$>	9 9 9	
\$> export HOLA="ls -l - a" \$> echo \$HOLA		ок	ок	ls -l - a \$>	9 9	
\$> export HOLA="s -la" \$> l\$HOLA		ОК	O.K	total 16 drwx 2 vietdu91 [] . drwxrwxewt 51 root []rwxr-xr-x 8 vietdu91 [] crashtest.c [] \$>	9	
<pre>\$> export HOLA="s -la" \$> l"\$HOLA"</pre>		ок	ок	bash: ls -la: command not found \$>	127	
<pre>\$> export HOLA="s -la" \$> l'\$HOLA'</pre>		ОК	ок	bash: l\$HOLA: command not found \$>	127	
<pre>\$> export HOLA="l" \$> \$HOLAs</pre>		ОК	ок		9	
<pre>\$> export HOLA="l" \$> "\$HOLA"s</pre>		ОК	ок	<pre>crashtest.c Docs Dockers Drawings Makefile minishell @ philosophers srcs test.c \$></pre>	9	
<pre>\$> export HOL=A=bonjour \$> env</pre>		ОК	ок	 HOL=A=bonjour \$>	9	
<pre>\$> export HOLA="l" \$> '\$HOLA's</pre>		ок	ок	bash: \$HOLAs: command not found \$>	127	
\$> export HOL=A="" \$> env		ОК	ок	 HOL=A= \$>	-	
\$> export TE+S=T \$> env		ок	ок	<pre>bash: export: `TE+S=T': not a valid identifier \$></pre>	1	
\$> export ""=""		ок	OK .	bash: export: `=': not a valid identifier 1 \$>	1	
\$> export ''=''		ОК	ок	<pre>bash: export: `=': not a valid identifier \$></pre>	1	
\$> export "="="="		ОК	ОК	<pre>bash: export: `===': not a valid identifier \$></pre>	1	

\$> export '='='='	ОК	ок	<pre>bash: export: `===': not a valid identifier \$></pre>	1	
<pre>\$> export HOLA=p \$> export BYE=w \$> \$HOLA"BYE"d</pre>	ок	ОК	bash: pBYEd: command not found \$>	127	
\$> export HOLA=p \$> export BYE=w \$> "\$HOLA"'\$BYE'd	ок		bash: p\$BYEd: command not found \$>	127	
\$> export HOLA=p \$> export BYE=w \$> "\$HOLA""\$BYE"d	ОК	ОК	/home/vietdu91/42_works/minishell	0	
\$> export HOLA=p \$> export BYE=w \$> \$"HOLA"\$"BYE"d	ОК	ОК	bash: HOLABYEd: command not found \$>	127	
\$> export HOLA=p \$> export BYE=w \$> \$'HOLA'\$'BYE'd	ОК		bash: HOLABYEd: command not found \$>	127	
<pre>\$> export HOLA=-n \$> "echo \$HOLA" hey</pre>	ОК	OK .	bash: echo -n: command not found \$>	127	
<pre>\$> export A=1 B=2 C=3 D=4 E=5 F=6 G=7 H=8 \$> echo "\$A'\$B"'\$C"\$D'\$E'"\$F'"'\$G'\$H"</pre>	ОК		1'2\$C"\$D5"\$F'7'8 \$>	0 0	
<pre>\$> export HOLA=bonjour \$> env \$> unset HOLA \$> env</pre>	ОК	ок	 PATH= \$>	0 0 0 0	
<pre>\$> export HOLA=bonjour \$> env \$> unset HOLA \$> unset HOLA \$> env</pre>	ОК		PATH= \$>	0 0 0 0 0	
\$> unset PATH \$> echo \$PATH	ОК	ОК	\$>	0	
\$> unset PATH \$> ls	ОК		bash: ls: No such file or directory	127	
\$> unset ""	ОК	UK	bash: unset: `': not a valid identifier	1	
\$> unset INEXISTANT	ОК		\$> \$>	0	Cas où la variable n'existe pas
\$> unset PWD \$> env grep PWD \$> pwd	ОК	ок	\$> \$> /home/vietdu91/42_works/minishell \$>	0 0 0	
<pre>\$> pwd \$> unset PWD \$> env grep PWD \$> cd \$PWD \$> pwd</pre>	ОК		/home/vietdu91/42_works/minishell \$> \$> \$> /home/vietdu91	0 0 0 0 0	
\$> unset OLDPWD \$> env grep OLDPWD	ОК	ОК	\$> \$>	0	
\$> unset 9HOLA	ОК	ОК	bash: unset: `9HOLA': not a valid identifier \$>	1	
\$> unset HOLA9	ОК		\$>	0	
\$> unset HOL?A	ОК	ОК	bash: unset: `HOL?A': not a valid identifier \$>	1	
\$> unset HOLA HOL?A	ОК		bash: unset: `HOL?A': not a valid identifier \$>	1	Il unset bien HOLA
\$> unset HOL?A HOLA	ОК		bash: unset: `HOL?A': not a valid identifier	1	Il unset bien HOLA
\$> unset HOL?A HOL.A	ОК	ОК	bash: unset: `HOL?A': not a valid identifier bash: unset: `HOL.A': not a valid identifier \$>	1	
\$> unset HOLA=	ОК		bash: unset: `HOLA=': not a valid identifier	1	
\$> unset HOL\\\\A NON GERE	ОК	ОК	bash: unset: `HOL\\A': not a valid identifier	1	
\$> unset HOL;A NON GERE	ОК	ОК	bash: A: command not found	127	
\$> unset HOL.A	ОК	ОК	bash: unset: `HOL.A': not a valid identifier	1	
\$> unset HOL+A	ОК		bash: unset: `HOL+A': not a valid identifier	1	
\$> unset HOL=A	ОК	- · · ·	bash: unset: `HOL=A': not a valid identifier	1	
\$> unset HOL{A	ОК		bash: unset: `HOL{A': not a valid identifier	1	
\$> unset HOL}A	ОК	ОК	bash: unset: `HOL}A': not a valid identifier	1	

	\$> unset HOL-A	Oł	ОК	bash: unset: `HOL-A': not a valid identifier	1	
	\$> unset -HOLA	Ol	ОК	bash: unset: -H: invalid option	2	
	\$> unset _HOLA	OH	ОК	\$>	0	
	\$> unset HOL_A	Ol	ОК	\$>	0	
	\$> unset HOLA_	OH	ОК	\$>	0	
	\$> unset HOL*A	Oł	~	bash: unset: `HOL*A': not a valid identifier	1	
	\$> unset HOL#A	OH		bash: unset: `HOL#A': not a valid identifier	1	
	\$> unset \$HOLA	OH	ок	\$>	0	Il va unset le contenu de \$HOLA, c'est-a-dire la variable bonjour
	\$> unset \$PWD	Oł	ОК	bash: unset: `/home/vietdu91/42_works/minishell': not a valid identifier	1	
	\$> unset HOL@	OH		bash: unset: `HOL@': not a valid identifier	1	
	\$> unset HOL!A	OH		bash: !A: event not found	1	
	\$> unset HOL^A	OH	ОК	bash: unset: `HOL^A': not a valid identifier	1	
	\$> unset HOL\$?A	OH	ок	\$>	0	Unset prend en compte le contenu de \$?, c'est-a-dire qu'il lit HOLOA ou HOL127A
	\$> unset HOL\~A	OH	ОК	bash: unset: `HOL~': not a valid identifier	1	
	\$> unset "" HOLA \$> env grep HOLA	OH		<pre>bash: unset: `': not a valid identifier \$> \$></pre>	1 0	HOLA sera quand meme unset !!
	\$> unset PATH	OH			0	
	\$> echo \$PATH \$> unset PATH	Oł		\$> bash: cat: No such file or directory	0	
	\$> cat Makefile		OK .	\$> bash: unset: `=': not a valid identifier	1	
	\$> unset = \$> unset ======	OH		bash: unset: `=': not a valid identifier bash: unset: `=====': not a valid identifier	0	
	\$> unset +++++	Ol		bash: unset: `+++++++': not a valid identifier	0	
	\$> unset	Oł Oł		\$>	0 0	
	\$> unset export	Ol		\$>	0	
	\$> unset echo	Oł		\$>	0	
	\$> unset pwd	OH		\$>	0	
	\$> unset cd	OH		*	0	
	\$> unset unset	OH		\$>	0	
	\$> unset sudo	OH	ОК	\$>	0	
	\$> export hola unset hola echo \$?	Oł	ОК	0 \$>	0	
	\$> /bin/echo	OF	OK	\$>	0	
RS S	\$> /bin/echo Hola Que Tal	OH	OK	Hola Que Tal	0	
# #	\$> /bin/env	J.		\$>	0	
FICE	Ç. Youngam	OH	ОК	PATH= \$>		
<u> </u>	\$> /bin/cd Desktop	OH	ОК	bash: /bin/cd: No such file or directory	127	
HISTORIQUE	\$> history NON GERE	OH	ОК	 558 echo \$? 559 clear 560 history \$>	0	Affiche les 500 dernières commandes
HIS.	\$> [touche du haut]	OH	ОК	\$> clear	1	Defile l'historique
	\$> pwd	Oł	ок	/home/vietdu91/42_works/minishell \$>	0	Retourne \$PWD
	\$> pwd hola	OH	ОК	/home/vietdu91/42_works/minishell \$>	0	
	\$> pwd ./hola	OH	ОК	/home/vietdu91/42_works/minishell \$>	0	
	\$> pwd hola que tal	OH	ОК	/home/vietdu91/42_works/minishell \$>	0	
	\$> pwd -p	Oł	ОК	bash: pwd: -p: invalid option \$>	2	
	\$> pwdp	Ol	ок	bash: pwd:: invalid option \$>	2	

	\$> pwdp		Oł	ОК	bash: pwd:: invalid option S>	2	
	\$> pwd p	NON GERE	OH	ОК	/home/vietdu91/42_works/minishell \$>	0	
	\$> pwd pwd pwd		Ok	ОК	/home/vietdu91/42_works/minishell \$>	0	N'affiche qu'une fois
	\$> pwd ls		Ok	ОК		0	
	\$> pwd ls env		Oł	ОК	/home/vietdu91/42_works/minishell	0	
	\$> cd		Oł	ОК	\$>	0	On doit se retrouver dans le dossier défini par \$HOME
	C. of				Α.		/home/vietdu91 L'user reste dans le même dossier
	\$> cd .		Ok		\$>		/home/vietdu91/42_works/minishell
	\$> cd ./		Ok		\$>		/home/vietdu91/42_works/minishell
	\$> cd ././.		Ok		\$>		/home/vietdu91/42_works/minishell
	\$> cd ./././		OF	ОК	\$>		/home/vietdu91/42_works/minishell
	\$> cd		Oł	ОК	\$>	U	On se retrouve au dossier précédent /home/vietdu91/42_works
	\$> cd/		OF	ОК	\$>	0	/home/vietdu91/42_works
	\$> cd/		OF	ОК	\$>	0	/home/vietdu91
	\$> cd/.		OF	ОК	\$>	0	/home/vietdu91/42_works
	\$> cd/././.		OF	ОК	\$>	0	/home/vietdu91/42_works
	\$> cd srcs		OF	ОК	\$>	0	/home/vietdu91/42_works/minishell/srcs
	\$> cd srcs objs		Ok	ОК	bash: cd: too many arguments \$>	1	/home/vietdu91/42_works/minishell
	\$> cd 'srcs'		OF	ОК	\$>	0	/home/vietdu91/42_works/minishell/srcs
	\$> cd "srcs"		OF	ОК	\$>	0	/home/vietdu91/42_works/minishell/srcs
	\$> cd '/etc'		Oł	ОК	\$>	0	Marche aussi avec les double quotes
	\$> cd /e'tc'		OF	ОК	\$>	0	/etc
	\$> cd /e"tc"		OF	ОК	\$>	0	/etc
	\$> cd sr		OF	ОК	bash: cd: sr: No such file or directory \$>	1	
	\$> cd Makefile		OF	ОК	bash: cd: Makefile: Not a directory \$>	1	
	<pre>\$> cd/minishell</pre>		OF	ОК	\$>	0	/home/vietdu91/42_works/minishell
	\$> cd////		Ok	ОК	\$>	0	/
	\$> cd/.///bin/ls		OF	ок	bash: cd:/.///bin/ls: No such file or directory \$>	1	/home/vietdu91/42_works/minishell
	\$> cd /		OF	ОК	\$>	0	1
	\$> cd '/'		OF	ОК	\$>	0	Marche aussi avec les double quotes
	\$> cd // \$> pwd	NON GERE	OH	ОК	// \$>	0	
	\$> cd '//' \$> pwd	NON GERE	Ok	ОК		0	Marche aussi avec les double quotes
	\$> cd /// \$> pwd		Ok	ОК	/ \$>	0	/
	\$> cd ////// \$> pwd		Ok	ОК	/ \$>	0	/
۵	\$> cd '/////' \$> pwd		Ok	ОК	<i> </i>	0	Marche aussi avec les double quotes
PWD	\$> cd /minishell		OH	ОК	bash: cd: /minishell: No such file or directory \$>	1	
	\$> cd / \$> cd		OH	ОК	<u>'</u>	0	1
CD	\$> cd _		Ok	ОК	\$>	1	
	\$> cd -		Oł	ОК	<pre>/home/vietdu91/42_works/minishell \$></pre>		Retourne à \$OLD_PWD \$OLD_PWD et \$PWD sont inversés /home/vietdu91/42_works/minishell

\$> cd	NON GERE		ОК	OK	\$>	0	Retourne à \$HOME \$OLD_PWD et \$PWD sont inversés /home/vietdu91
\$> cd			ОК	OK	bash: cd:: invalid option \$>	2	,,
\$> cd \$HOME			ОК	OK	\$>	0	/home/vietdu91
\$> cd \$HOME \$HOME			ОК	ОК	bash: cd: too many arguments \$>	1	
\$> cd \$HOME/42_works			ОК	ОК	\$>	0	/home/vietdu91/42_works
<pre>\$> cd "\$PWD/srcs"</pre>			ОК	ОК	\$>	0	/home/vietdu91/42_works/minishell/srcs
<pre>\$> cd '\$PWD/srcs'</pre>			ок	ОК	<pre>bash: cd: \$PWD/srcs: No such file or directory \$></pre>	1	
<pre>\$> unset HOME \$> cd \$HOME</pre>			ок	ОК	bash: cd: HOME not set \$>	1	
<pre>\$> unset HOME \$> export HOME= \$> cd</pre>			ок	ОК	\$>	0	/home/vietdu91/42_works/minishell
\$> unset HOME \$> export HOME \$> cd			ок	ОК	bash: cd: HOME not set \$>	1	
<pre>\$> cd minishell Docs crashtest.c</pre>			ОК	ОК	bash: cd: too many arguments \$>	1	
\$> cd / echo \$? \$> pwd			ОК	ОК	0 \$>	0	/home/vietdu91/42_works/minishell
\$> cd ~			ОК	OK	\$>	0	/home/vietdu91
\$> cd ~/	NON GERE		ок	ОК	\$>	0	/home/vietdu91
\$> cd ~/ echo \$? \$> pwd	NON GERE		ок	ОК	0 \$>	0	/home/vietdu91/42_works/minishell
\$> cd *	NON GERE		ОК	OK	\$>	0	[BONUS] S'il n'y a qu'un seul <u>dossier</u> et rien d'autre
\$> cd *	NON GERE		ОК	ОК	bash: cd: minishell: No such file or directory \$>	1	[BONUS] S'il n'y a qu'un seul <u>fichier</u> et rien d'autre
\$> cd *	NON GERE		ОК	ОК	bash: cd: too many arguments	1	[BONUS] S'il y au moins deux éléments
<pre>\$> mkdir a \$> mkdir a/b \$> cd a/b \$> rm -r//a \$> cd</pre>			ОК	ОК	chdir: error retrieving current directory: getcwd: cannot access parent directories: No such file or directory \$>	1	
<pre>\$> mkdir a \$> mkdir a/b \$> cd a/b \$> rm -r//a \$> pwd</pre>			ОК	ОК	<pre>pwd: error retrieving current directory: getcwd: cannot access parent directories: No such file or directory \$></pre>	1	
<pre>\$> mkdir a \$> mkdir a/b \$> cd a/b \$> rm -r//a \$> echo \$PWD \$> echo \$OLDPWD</pre>			ОК	ОК	/home/vietdu91/42_works/minishell/a/b \$> /home/vietdu91/42_works/minishell \$>	0	
<pre>\$> mkdir a \$> mkdir a/b \$> cd a/b \$> rm -r//a \$> cd \$> echo \$PWD \$> echo \$OLDPWD</pre>			ок	ОК	/home \$> /home/vietdu91/42_works/minishell/a/b \$>	0 0	
<pre>\$> mkdir a \$> cd a \$> rm -r/a \$> echo \$PWD \$> echo \$OLDPWD</pre>			ОК	ОК	<pre>//home/vietdu91/42_works/minishell/a \$>//home/vietdu91/42_works/minishell \$></pre>	0 0	
<pre>\$> export CDPATH=/ \$> cd \$HOME/</pre>	NON GERE		ок	ОК	\$>	0	A gerer ? /home
<pre>\$> export CDPATH=/ \$> cd home/vietdu91</pre>	NON GERE		ок	ОК	/home/vietdu91 \$>	0	A gerer ? /home/vietdu91
\$> export CDPATH=./ \$> cd .	NON GERE		ОК	ОК	\$>	0	A gerer ? /home/vietdu91/42_works/minishell
\$> export CDPATH=./ \$> cd	NON GERE		ОК	ОК	\$>	0	A gerer ? /home/vietdu91/42_works
\$> chmod 000 minishell \$> ./minishell			ОК	ОК	bash: ./minishell : Permission denied \$>	126	, mana, recedes 1, 12_not to

	\$> ls hola		ОК		ls: cannot access 'hola': No such file or directory \$>	2	
	\$> ./Makefile		ОК		bash: ./Makefile: Permission denied \$>	126	
	\$> ./minishell		ОК	ОК	\$>	0	Relance le Minishell Il faut donc 2 fois exit pour exit
	\$> env grep SHLVL \$> ./minishell				\$>	0	
	<pre>\$> env grep SHLVL \$> exit</pre>	NON GERE	ОК	ОК		0	
	<pre>\$> env grep SHLVL \$> touch hola \$> ./hola</pre>		ОК		bash: ./hola: Permission denied	126	
10	\$> ./nota \$> env "wc" -l		ОК		\$> 66 \$>	0	
BÂTARDS	\$> env "wc "-l		ОК		bash: wc -l: command not found \$>	127	
BÂT	\$> expr 1 + 1		ОК	ОК	2 \$>	0	
	\$> expr \$? + \$?		ок	OK	0 \$>	0	Fait la somme de la valeur de \$?
	<pre>\$> env -i ./minishell \$> env</pre>		ОК	OK	PWD=/mnt/nfs/homes/emtran/Desktop/42_minishell SHLVL=1 _=/usr/bin/env	0	
	\$> env -i ./minishell				\$> export OLDPWD	0	
	\$> export		ОК	ОК	<pre>export PWD="/mnt/nfs/homes/emtran/Desktop/42_minishell" export SHLVL="1" \$></pre>		
	<pre>\$> env -i ./minishell \$> cd</pre>		ОК		bash: cd: HOME not set \$>	1	
	<pre>\$> env -i ./minishell \$> cd ~</pre>	NON GERE	ОК	OK .	\$>	0	/home/vietdu91
	\$> exit		ОК		exit ~/minishell \$>	0	
	\$> exit exit		ОК	ОК	exit bash: exit: exit: numeric argument required ~/minishell \$>	2	
	\$> exit hola		ОК	ОК	exit bash: exit: hola: numeric argument required ~/minishell \$>	2	
	<pre>\$> exit hola que tal</pre>		ОК		exit bash: exit: hola: numeric argument required ~/minishell \$>	2	
	\$> exit 42		ОК	O.K	exit ~/minishell \$>	42	
	\$> exit 000042		ОК	ОК	exit ~/minishell \$>	42	
	\$> exit 666		ОК		exit ~/minishell \$>	154	
	\$> exit 666 666		ОК	ОК	exit bash: exit: too many arguments \$>	1	
	\$> exit -666 666		ок	ОК	<pre>exit bash: exit: too many arguments \$></pre>	1	
	\$> exit hola 666		ОК	OK	exit bash: exit: hola: numeric argument required ~/minishell \$>	2	
	\$> exit 666 666 666 666		ОК	ОК	exit bash: exit: too many arguments \$>	1	
	\$> exit 666 hola 666		ОК	ок	exit bash: exit: too many arguments	1	
	\$> exit hola 666 666		ОК	ОК	exit bash: exit: hola: numeric argument required ~/minishell \$>	2	
	\$> exit 259		ок	ОК	exit ~/minishell \$>	3	
	\$> exit -4		ОК	OK .	exit ~/minishell \$>	252	
	\$> exit -42		ОК	ОК	exit ~/minishell \$>	214	

	\$> exit -0000042	ОК	ОК	exit ~/minishell \$>	214	
	\$> exit -259	ОК	ОК	exit ~/minishell \$>	253	
	\$> exit -666	ОК	ОК	exit ~/minishell \$>	102	
	\$> exit +666	ОК	ОК	exit ~/minishell \$>	154	
	\$> exit 0	ОК	ОК	exit ~/minishell \$>	0	
	\$> exit +0	ОК	ОК	exit ~/minishell \$>	0	
	\$> exit -0	ОК	ОК	exit ~/minishell \$>	0	
_	\$> exit +42	ОК	ок	exit ~/minishell \$>	42	
EXI	\$> exit -69 -96	ОК	ОК	exit bash: exit: too many arguments \$>	1	
	\$> exit666	ОК	ОК	exit bash: exit:666: numeric argument required ~/minishell \$>	2	
	\$> exit ++++666	ОК	ОК	exit bash: exit: ++++666: numeric argument required ~/minishell \$>	2	
	\$> exit ++++++0	ок	ОК	exit bash: exit: ++++++0: numeric argument required ~/minishell \$>	2	
	\$> exit0	ОК	ОК	exit bash: exit:0: numeric argument required ~/minishell \$>	2	
	\$> exit "666"	ОК	OK .	exit ~/minishell \$>	154	
	\$> exit '666'	ОК	OK .	exit ~/minishell \$>	154	
	\$> exit '-666'	ОК	OK .	exit ~/minishell \$>		Marche aussi pour les double quotes
	\$> exit '+666'	ОК	OK .	exit ~/minishell \$>		Marche aussi pour les double quotes
	\$> exit '666'	ок	OK	exit bash: exit:666: numeric argument required ~/minishell \$>		Marche aussi pour les double quotes
	\$> exit '++++666'	ОК	ОК	exit bash: exit: ++++666: numeric argument required ~/minishell \$>		Marche aussi pour les double quotes
	\$> exit '6'66	ОК	OK .	exit ~/minishell \$>		Marche aussi pour les double quotes
	\$> exit '2'66'32'	ОК	OK .	exit ~/minishell \$>		Marche aussi pour les double quotes
	\$> exit "'666'"	ок	ОК	exit bash: exit: '666': numeric argument required ~/minishell \$>	2	
	\$> exit '"666"'	ОК	OK	exit bash: exit: "666": numeric argument required ~/minishell \$>	2	
	\$> exit '666'"666"666	ОК	OK .	exit ~/minishell \$>	170	
	\$> exit +'666'"666"666	ОК	OK .	exit ~/minishell \$>	170	
	\$> exit -'666'"666"666	ОК	OK .	exit ~/minishell \$>	86	
	\$> exit 9223372036854775807	ОК	OK .	exit ~/minishell \$>	255	
	\$> exit 9223372036854775808	ОК	ОК	exit bash: exit: 9223372036854775808: numeric argument required ~/minishell \$>		
	\$> exit -9223372036854775808	ОК	OK .	exit ~/minishell \$>	0	
	\$> exit -9223372036854775809	ОК	OK	exit bash: exit: -9223372036854775809: numeric argument required ~/minishell \$>	2	

	\$> cat cat cat ls				crashtest.c Docs Dockers Drawings Makefile minishell	0	Apparaît la ligne ls
			0.14		philosophers srcs test.c		Et 3 retours à la ligne avec la touche Entrée
			ОК	OK			
					\$>		
	\$> ls exit				\$>	0	Ne fait rien
			ОК	ОК			Chaque commande d'un pipeline est exécutée dans son propre sous-shell, qui est un processus
	\$> ls exit 42		014	01/	6.	42	distinct Ne fait rien
	\$> exit ls		ОК	ОК	<pre>\$> crashtest.c Docs Dockers Drawings Makefile minishell</pre>		Ne Tall Flen
	3> EXCC C3		ОК	ОК	philosophers srcs test.c \$>	o .	
	\$> echo hola > bonjour \$> exit cat -e bonjour		ОК	ОК	hola\$ \$>	0	
	<pre>\$> echo hola > bonjour \$> cat -e bonjour exit</pre>		ОК	ОК	\$>	0	
	\$> echo echo		ок	ОК	\$>	0	
	\$> echo hola echo que tal		ОК	ок	que tal \$>	0	
	\$> pwd echo hola		ОК	ОК	hola \$>	0	
	\$> env echo hola		ОК		hola \$>	0	
	\$> echo oui cat -e		ОК	ок	oui\$ \$>	0	
	\$> echo oui echo non echo hola grep oui		ОК	ОК	\$>	0	
	\$> echo oui echo non echo hola grep non		ОК	ОК	\$>	0	
	\$> echo oui echo non echo hola grep hola		ОК	ОК	hola \$>	0	
	\$> echo hola cat -e cat -e cat -e		ОК	ОК	hola\$\$\$ \$>	0	
	\$> cd echo "hola"		ОК	ОК	hola \$>	0	/home/vietdu91/42_works/minishell
	\$> cd / echo "hola"		ОК	ОК	hola \$>	0	/home/vietdu91/42_works/minishell
	\$> cd pwd		ОК	ОК	/home/vietdu91/42_works/minishell \$>	0	/home/vietdu91/42_works/minishell
	<pre>\$> ifconfig grep ":"</pre>		ОК	OK	docker0: flags=4099 <up,broadcast,multicast> mtu 1500 ether 02:42:27:cf:95:49 txqueuelen 0 (Ethernet)</up,broadcast,multicast>	0	
					[] \$>		
	\$> ifconfig grep hola		ОК	ОК	\$>	1	
	\$> whoami grep \$USER		ОК	OK .	vietdu91 \$>	0	
	<pre>\$> whoami grep \$USER > /tmp/bonjour \$> cat /tmp/bonjour</pre>		ОК	ОК	vietdu91 \$>	0	
	<pre>\$> whoami cat -e cat -e > /tmp/bonjour \$> cat /tmp/bonjour</pre>		ОК	ок	<pre>vietdu91\$\$ \$></pre>	0	
	<pre>\$> whereis ls cat -e cat -e > /tmp/bonjour \$> cat /tmp/bonjour</pre>		ОК	ОК	ls: /usr/bin/ls /usr/share/man/man1/ls.1posix.gz /usr/share/man/man1/ls.1.gz\$\$ \$>	0	
	\$> ls hola		ОК	ОК	bash: hola: command not found \$>	127	
PIPES	\$> ls ls hola		ОК	ОК	ls: cannot access 'hola': No such file or directory \$>	2	
	\$> ls ls hola		ОК	ОК	bash: hola: command not found \$>	127	
	\$> ls hola ls		ОК	ОК	bash: hola: command not found crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$>	0	
	\$> ls ls hola rev		ОК	OK .	bash: hola: command not found \$>	0	
	\$> ls ls echo hola rev		ОК	ОК	aloh \$>	0	

\$> ls -la grep "."				total 16 drwx 2 vietdu91 [] .	0	Sort la commande entiere de ls -la
		ОК	ОК	drwxrwxewt 51 root []rwxr-xr-x 8 vietdu91 [] crashtest.c		
				[] \$>		
\$> ls -la grep "'.'"		ОК	ОК	\$>	0	
<pre>\$> echo test.c cat -e cat -e cat -e cat - e cat -e cat -e cat -e cat -e cat - e cat -e</pre>		ОК	ОК	test.c\$	0	
\$> ls l		ок	OK	crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$>	0	
<pre>\$> echo hola cat cat cat cat grep hola</pre>		ОК	ОК	hola \$>	0	
\$> echo hola cat		ОК	ОК	hola \$>	0	
\$> echo hola cat		ОК	OK	hola \$>	0	
\$> echo hola cat		ОК		hola \$>	0	
\$> echo hola cat		ОК	ОК	hola \$>	0	
\$> echo hola cat NON GERE		ОК		hola \$>	0	[BONUS]
\$> echo hola cat		ОК	OK	bash: syntax error near unexpected token ` ' \$>	2	
\$> ech o hola cat		ОК		bash: ech: command not found bash: o: command not found \$>	127	
\$> cat Makefile cat -e cat -e		ОК	ОК	[] \$\$.PHONY: all clean fclean re\$\$ \$>	0	
\$> cat Makefile grep srcs cat -e		ОК	ОК	objs/%.o: srcs/%.c\$ @norminette srcs/*\$ \$>	0	
<pre>\$> cat Makefile grep srcs grep srcs cat - e</pre>		ОК	ОК	objs/%.o: srcs/%.c\$ @norminette srcs/*\$ \$>	0	
<pre>\$> cat Makefile grep pr head -n 5 cd file_not_exist</pre>		ОК	ОК	<pre>bash: cd: file_not_exist: No such file or directory \$></pre>	1	
\$> cat Makefile grep pr head -n 5 hello		ОК	ОК	bash: hello: command not found \$>	127	
\$> export HOLA=bonjour cat -e cat -e		ОК	ОК	\$>	0	
\$> unset HOLA cat -e		ОК		\$>	0	
<pre>\$> export HOLA echo hola \$> env grep PROUT</pre>		ОК	ОК	hola \$> \$>	0	
\$> export echo hola		ОК	ОК	hola \$>	0	
\$> sleep 3 sleep 3		ОК	ОК	\$>	0	Le prompt apparait au bout de 3 secondes
\$> time sleep 3 sleep 3		ОК	OK	real0m3.002s user0m0.003s sys0m0.001s \$>	0	Le prompt apparait au bout de 3 secondes
\$> sleep 3 exit		OK	ОК	\$>	0	Le prompt apparait au bout de 3 secondes
\$> exit sleep 3		OK		\$>	0	Le prompt apparait au bout de 3 secondes
<pre>\$> echo hola > a \$> >>b echo que tal \$> cat a <b cat="" =""> c cat</pre>		ок		\$>	0	
\$> pwd && ls NON GERE		ОК	ОК	/home/vietdu91/42_works/minishell crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$>	0	[BONUS]
\$> pwd ls NON GERE		ОК	ОК	/home/vietdu91/42_works/minishell \$>	0	[BONUS]
\$> echo hola echo bonjour NON GERE		ОК	ОК	hola \$>	0	[BONUS]

	\$> echo hola && echo bonjour	NON GERE	ОК	ОК	hola bonjour \$>	0	[BONUS]
	\$> echo bonjour echo hola	NON GERE	ОК	ОК	bonjour \$>	0	[BONUS]
	\$> echo bonjour && echo hola	NON GERE	ОК	ОК	bonjour hola \$>	0	[BONUS]
	\$> echo -n bonjour && echo -n hola	NON GERE	ОК	ОК	bonjourhola\$>	0	[BONUS]
	\$> pwd && ls && echo hola	NON GERE	ок		/home/vietdu91/42_works/minishell crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c hola \$>	0	[BONUS]
	\$> pwd ls && echo hola	NON GERE	ОК	ок	/home/vietdu91/42_works/minishell hola \$>	0	[BONUS]
	\$> pwd && ls echo hola	NON GERE	ОК	ОК	/home/vietdu91/42_works/minishell crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$>	0	[BONUS]
	\$> pwd ls echo hola	NON GERE	ОК	ОК	/home/vietdu91/42_works/minishell \$>	0	[BONUS]
	\$> ls export ""	NON GERE	ОК	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$></pre>	0	[BONUS]
=	\$> export "" ls	NON GERE	ОК	ОК	bash: export: `': not a valid identifier crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$>	0	[BONUS]
& &	\$> ls && export ""	NON GERE	ОК	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c bash: export: `': not a valid identifier \$></pre>	1	[BONUS]
	\$> export "" && ls	NON GERE	ОК		<pre>bash: export: `': not a valid identifier \$></pre>	1	[BONUS]
	\$> ls ;	NON GERE	ОК	ОК	bash: syntax error near unexpected token `;' \$>	2	[BONUS] Marche pour les metacharacters &;()
	\$> ; ls	NON GERE	ОК	ОК	<pre>bash: syntax error near unexpected token `;' \$></pre>	2	[BONUS] Marche pour les metacharacters &;()
	\$> ls && ;	NON GERE	ОК	ОК	<pre>bash: syntax error near unexpected token `;' \$></pre>	2	[BONUS] Marche pour les metacharacters &;()
	\$> ; && ls	NON GERE	ОК	OK	<pre>bash: syntax error near unexpected token `;' \$></pre>	2	[BONUS] Marche pour les metacharacters &;()
	\$> ls <	NON GERE	ОК	ОК	<pre>bash: syntax error near unexpected token `newline' \$></pre>	2	[BONUS] Marche pour les metacharacters <>
	\$> ls && <	NON GERE	ОК	ок	<pre>bash: syntax error near unexpected token `newline' \$></pre>	2	[BONUS] Marche pour les metacharacters <>
	\$> cat echo ls	NON GERE	ок	ОК	\$>	0	[BONUS]
	\$> cat echo && ls	NON GERE	ОК	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$></pre>	0	[BONUS]
	\$> ls cat echo	NON GERE	ОК	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$></pre>	0	[BONUS]
	\$> ls && cat echo	NON GERE	ОК	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$></pre>	0	[BONUS]
	\$> export "" && unset ""	NON GERE	ОК	O.K	bash: export: `': not a valid identifier	1	[BONUS]
	\$> (ls)	NON GERE	ОК	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$></pre>		[BONUS]
	\$> ((ls))	NON GERE	ОК		<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$></pre>	0	[BONUS]

\$> (() ls)	NON GERE	ОК	OK	bash: syntax error near unexpected token `)' 2 \$>	[BONUS]
\$> ls && (ls)	NON GERE	ок	ок	crashtest.c Docs Dockers Drawings Makefile minishell 0 philosophers srcs test.c crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$>	[BONUS]
\$> (ls && pwd)	NON GERE	ок	ОК	crashtest.c Docs Dockers Drawings Makefile minishell 0 philosophers srcs test.c /home/vietdu91/42_works/minishell	[BONUS]
\$> ((ls&&pwd))	NON GERE	ок	ОК	crashtest.c Docs Dockers Drawings Makefile minishell 0 philosophers srcs test.c /home/vietdu91/42_works/minishell \$\cdots\$	[BONUS]
\$> ((ls) &&pwd)	NON GERE	ок	ОК	crashtest.c Docs Dockers Drawings Makefile minishell 0 philosophers srcs test.c /home/vietdu91/42_works/minishell \$>	[BONUS]
\$> (ls && ((pwd)))	NON GERE	ок	ОК	crashtest.c Docs Dockers Drawings Makefile minishell 0 philosophers srcs test.c /home/vietdu91/42_works/minishell S>	[BONUS]
\$> (ls && pwd) > hola \$> cat hola	NON GERE	ОК	ОК	crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c /home/vietdu91/42_works/minishell \$>	[BONUS]
\$> > hola ls && pwd	NON GERE	ОК	ОК	/home/vietdu91/42_works/minishell 0	[BONUS]
\$> > hola (ls && pwd)	NON GERE	ОК	ОК	bash: syntax error near unexpected token `(' 2 \$>	[BONUS]
\$> (> pwd) \$> ls	NON GERE	ок	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers pwd srcs test.c \$></pre>	[BONUS]
\$> (< pwd) \$> ls	NON GERE	ОК	ОК	\$> 0	[BONUS] Si pwd existe
\$> (< pwd)	NON GERE	ОК	ОК	bash: pwd: No such file or directory \$>	[BONUS] Si pwd n'existe pas
\$> ((((pwd))))	NON GERE	ОК	ОК	/home/vietdu91/42_works/minishell 0	[BONUS]
\$> () pwd	NON GERE	ОК	ОК	bash: syntax error near unexpected token `)' \$>	[BONUS]
\$> > pwd (ls)	NON GERE	ОК	ОК	bash: syntax error near unexpected token `(' 2 \$>	[BONUS]
\$> (ls pwd)&&(ls pwd)	NON GERE	ОК	ок	crashtest.c Docs Dockers Drawings Makefile minishell 0 philosophers pwd srcs test.c crashtest.c Docs Dockers Drawings Makefile minishell philosophers pwd srcs test.c \$>	[BONUS]
\$> (lss pwd)&&(lss pwd)	NON GERE	ОК	ОК	bash: lss: command not found /home/vietdu91/42_works/minishell bash: lss: command not found /home/vietdu91/42_works/minishell \$>	[BONUS]
\$> (lss&&pwd)&&(lss&&pwd)	NON GERE	ОК	OK	bash: lss: command not found 127	[BONUS]
\$> (ls && pwd wc) > hola \$> cat hola	NON GERE	ОК	ОК	crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c 1 1 34 \$>	[BONUS]

ш		_					
(PARENTHESI	<pre>\$> (ls && pwd wc) > hola \$> (ls && pwd wc) > hola \$> cat hola</pre>	NON GERE	ОК		<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c</pre>	0 0	[BONUS]
	<pre>\$> (ls && pwd wc) >> hola \$> echo hey&&(ls && pwd wc) > hola \$> cat hola</pre>	NON GERE	ОК	ОК	<pre>hey \$> crashtest.c Docs [] srcs test.c</pre>	0 0 0	[BONUS]
	\$> (pwd wc) < hola	NON GERE	ОК	ок	1 1 34 \$>	0	[BONUS]
	\$> (ls && pwd wc) < hola	NON GERE	ОК	ок	crashtest.c Docs Dockers Drawings Makefile minishell philosophers pwd srcs test.c 1 34	0	[BONUS]
	\$> (ls -z pwd wc) < hola	NON GERE	ОК	ОК	ls: invalid option 'z' 1 34 \$>	0	[BONUS]
	<pre>\$> echo hey > hola \$> (pwd wc) < hola</pre>	NON GERE	ОК	ОК	1 1 34 \$>	0	[BONUS]
	<pre>\$> echo hey > hola \$> (ls && pwd wc) < hola</pre>	NON GERE	ОК	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers pwd srcs test.c</pre>	0	[BONUS]
	\$> echo hey > hola \$> (ls -z pwd wc) < hola	NON GERE	ок	ОК	ls: invalid option 'z' 1 1 34 \$>	0	[BONUS]
	\$> (ls -z pwd && ls)	NON GERE	ОК	OK	ls: invalid option 'z' /home/vietdu91/42_works/minishell crashtest.c Docs Dockers Drawings Makefile minishell philosophers pwd srcs test.c \$>	0	[BONUS]
	\$> ls (cat Makefile grep srcs) && (pwd wc)	NON GERE	ОК	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers pwd srcs test.c</pre>	0	[BONUS]
	\$> ls -z && (ls) && (pwd)	NON GERE	ОК		ls: invalid option 'z' \$>	130	[BONUS]
	<pre>\$> (ls > Docs/hey && pwd) > hola \$> cat hola \$> cat Docs/hey</pre>	NON GERE	ок		/home/vietdu91/42_works/minishell \$> crashtest.c Docs [] srcs test.c \$>	0 0 0	[BONUS]
	<pre>\$> ls > Docs/hey && pwd > hola \$> cat hola \$> cat Docs/hey</pre>	NON GERE	ок	ОК	/home/vietdu91/42_works/minishell \$> crashtest.c Docs [] srcs test.c \$>	0 0 0	[BONUS]
	\$> cd/ && pwd && pwd	NON GERE	ОК		/home/vietdu91/ /home/vietdu91/ \$>	0	[BONUS]
	\$> (cd/ && pwd) && pwd	NON GERE	ОК	ОК	/home/vietdu91/ /home/vietdu91/42_works/minishell \$>	0	[BONUS]

\$> ls -z cd//&&	pwd				ls: invalid option 'z'	0	[BONUS]
\$> pwd	NON GE	RE	ОК	ок	<pre>/home/vietdu91 \$> /home/vietdu91</pre>	0	
\$> ls -z (cd//&	2 oud)				\$> ls: invalid option 'z'	0	[BONUS]
\$> pwd	NON GE	RF	ОК	ок	/home/vietdu91 \$>	0	[BOUG2]
	11011 02	_	O.C		/home/vietdu91/42_works/minishell \$>		
<pre>\$> echo hola > bonjour \$> cat bonjour</pre>			ОК	ОК	hola \$>	0	
<pre>\$> echo que tal >> bonjou \$> cat bonjour</pre>	Г		ОК	ок	hola que tal	0	
					\$ > \$>	0	
<pre>\$> echo hola > bonjour \$> echo que tal >> bonjou \$> cat < bonjour</pre>	г		OK	ок	\$> hola	0 0	
					que tal \$>		
<pre>\$> echo hola > bonjour \$> rm bonjour</pre>			ОК	ок	que tal \$>	0	
<pre>\$> echo que tal >> bonjou \$> cat < bonjour</pre>						0	
<pre>\$> echo hola que tal > bo \$> cat bonjour</pre>			ОК	OK .	hola que tal \$>	0	
<pre>\$> echo hola que tal > /t \$> cat -e /tmp/bonjour</pre>	mp/bonjour		ОК	ок	hola que tal\$ \$>	0	
<pre>\$> export HOLA=hey \$> echo bonjour > \$HOLA \$> echo \$HOLA</pre>			ОК	ОК	\$> hola \$>	0 0 0	Ne fait rien
\$> whereis grep > Docs/bo \$> cat Docs/bonjour	njour			2	grep: /usr/bin/grep /usr/share/man/man1/grep.1.gz /usr/share/man/man1/grep.1posix.gz /usr/share/info/grep.	0	
Ç cae boes/bonjou			ОК	ОК	info.gz \$>		
<pre>\$> ls -la > Docs/bonjour \$> cat Docs/bonjour</pre>					total 16	0	
			ОК	ок	drwx 2 vietdu91 [] . drwxrwxewt 51 root []rwxr-xr-x 8 vietdu91 [] crashtest.c		
					[] \$>		
<pre>\$> pwd>bonjour \$> cat bonjour</pre>			ОК	ок	<pre>/home/vietdu91/42_works/minishell \$></pre>	0	
<pre>\$> pwd > \$> cat bonjour</pre>	bonjour		ОК	ок	/home/vietdu91/42_works/minishell \$>	0	
<pre>\$> echo hola > > bonjour</pre>			ок	ОК	bash: syntax error near unexpected token `>' \$>	2	
<pre>\$> echo hola < < bonjour</pre>			ОК	ОК	bash: syntax error near unexpected token `<' \$>	2	
<pre>\$> echo hola >>> bonjour</pre>			ОК	ОК	bash: syntax error near unexpected token `>' \$>	2	
<pre>\$> > bonjour echo hola \$> cat bonjour</pre>			ОК	ок	hola \$>	0 0	
\$> > bonjour echo hola \$> cat bonjour			ОК	ОК	hola \$> \$>	0 0	
<pre>\$> prout hola > bonjour \$> ls</pre>			ОК	ОК	bash: prout: command not found \$>	127 0	Erreur mais file "bonjour" cree tout de meme
\$> echo hola > hello >> h	ello >> hello				bonjour [] \$> hello	0	
<pre>\$> ls \$> cat hello</pre>			ок	ОК	hello \$> hola	0	
\$> echo hola > hello >> h	alla xx halla				\$> hola		
<pre>\$> echo hola > hello >> h \$> echo hola >> hello \$> cat < hello</pre>	etto->> netto		ОК	ОК	nola hola \$>	0 0	
\$> echo hola > hello >> h \$> echo hola >> hello	ello >> hello				hola \$>	0	
<pre>\$> echo hota >> hetto \$> echo hola > hello >> h \$> cat < hello</pre>	ello >> hello		ОК	ОК		0	
\$> echo hola >> hello >> \$> echo hola >> hello	hello > hello		ОК	ОК	hola hola	0 0	
\$> cat < hello				- OK	\$>	0	

<pre>\$> echo hola >> hello >> hello \$> echo hola >> hello \$> echo hola >> hello >> hello \$> cat < hello</pre>		ОК	ОК	hola \$> 0 0 0 0
<pre>\$> echo hola > hello \$> echo hola >> hello >> hello \$> echo hola >> hello \$> cat < hello</pre>		ОК	ОК	hola
<pre>\$> echo hola > hello \$> echo hey > bonjour \$> echo <bonjour <hello<="" pre=""></bonjour></pre>		ОК	ОК	\$> 0 0 0
<pre>\$> echo hola > hello \$> echo hey > bonjour \$> echo <hello <bonjour<="" pre=""></hello></pre>		ОК	ОК	\$> 0 0 0 0
<pre>\$> echo hola > bonjour \$> echo hey > hello \$> rm bonjour hello \$> echo hola > bonjour > hello > bonjour \$> cat bonjour \$> cat hello</pre>		ОК	ОК	hola
<pre>\$> echo hola > bonjour \$> echo hey > hello \$> echo hola > bonjour > hello > bonjour \$> cat bonjour \$> cat hello</pre>		ОК	ОК	hola
<pre>\$> echo hola > bonjour \$> echo hey > hello \$> rm bonjour hello \$> echo hola > bonjour >> hello > bonjour \$> cat bonjour \$> cat hello</pre>		ОК	ОК	hola
<pre>\$> echo hola > bonjour \$> echo hey > hello \$> echo hola > bonjour > hello > bonjour \$> cat bonjour \$> cat hello</pre>		ОК	ОК	hola \$> \$> 0 0 0 0
<pre>\$> echo hola > bonjour \$> echo hey > hello \$> rm bonjour hello \$> echo hola > bonjour > hello >> bonjour \$> cat bonjour \$> cat hello</pre>		ОК	ОК	hola
<pre>\$> echo hola > bonjour \$> echo hey > hello \$> echo hola > bonjour > hello >> bonjour \$> cat bonjour \$> cat hello</pre>		ОК	ОК	hola
<pre>\$> echo hola > bonjour \$> echo hey > hello \$> rm bonjour hello \$> echo hola >> bonjour > hello > bonjour \$> cat bonjour \$> cat hello</pre>		ОК		hola
<pre>\$> echo hola > bonjour \$> echo hey > hello \$> echo hola >> bonjour > hello > bonjour \$> cat bonjour \$> cat hello</pre>		ОК	ОК	hola \$> \$> 0 0 0 0
<pre>\$> echo hola > bonjour \$> echo hey > hello \$> rm bonjour hello \$> echo hola >> bonjour >> hello >> bonjour \$> cat bonjour \$> cat hello</pre>		ОК	ОК	hola
<pre>\$> echo hola > bonjour \$> echo hey > hello \$> echo hola >> bonjour >> hello >> bonjour \$> cat bonjour \$> cat hello</pre>		ОК	ОК	hola
<pre>\$> > bonjour echo hola bonjour \$> cat bonjour</pre>		ОК	ОК	bonjour hola
<pre>\$> >bonjour echo > hola>bonjour>hola>>bonjour>hola hey >bonjour hola >hola \$> cat bonjour \$> cat hola</pre>		ОК	ОК	\$>

	<pre>\$> echo bonjour > hola1 \$> echo hello > hola2 \$> echo 2 >hola1 >> hola2 \$> ls \$> cat hola1</pre>		ОК		hola1 hola2 [] \$> \$> helo	0 0 0	
	\$> cat hotal \$> cat hota2				z \$>	0	
	<pre>\$> echo bonjour > hola1 \$> echo hello > hola2 \$> echo 2 >>hola1 > hola2 \$> ls \$> cat hola1 \$> cat hola2</pre>		ОК		hola1 hola2 [] \$> bonjour \$> 2	0 0 0 0	
	\$> > pwd		ОК	ОК	pwd	0	
	\$> ls				\$> bash: pwd: No such file or directory	0	
			ОК	OK .	<pre>\$> .: not enough arguments</pre>	1	Voir pourquoi
	I N	ION GERE	ОК	OK .	\$>	1	voti pour quot
	\$> cat <pwd< td=""><td></td><td>ОК</td><td>ОК</td><td>bash: pwd: No such file or directory \$></td><td>1</td><td></td></pwd<>		ОК	ОК	bash: pwd: No such file or directory \$>	1	
	\$> cat <srcs pwd<="" td=""><td></td><td>ОК</td><td>ОК</td><td>bash: srcs/pwd: No such file or directory \$></td><td>1</td><td></td></srcs>		ОК	ОК	bash: srcs/pwd: No such file or directory \$>	1	
	\$> cat <td></td> <td>ОК</td> <td>ОК</td> <td>bash:/pwd: No such file or directory S></td> <td>1</td> <td></td>		ОК	ОК	bash:/pwd: No such file or directory S>	1	
	\$> cat >>		ОК	ОК	bash: syntax error near unexpected token `newline' 5>	2	Marche aussi pour les autres operateurs de redirection
	\$> cat >>>		ОК	ОК	bash: syntax error near unexpected token `>'	2	redi ección
^	\$> cat >> <<		ОК	ОК	bash: syntax error near unexpected token `<<'	2	
V V	\$> cat >> >> << >>		ОК	ОК	bash: syntax error near unexpected token `>' \$>	2	
^ ^	\$> cat < ls		ОК	ОК	bash: ls: No such file or directory	1	
^ ^	\$> cat < ls > ls		ОК	ОК	bash: ls: No such file or directory	1	
^ ^	\$> cat > ls1 < ls2				\$> bash: ls2: No such file or directory	1	
^	\$> ls		ОК	ОК	\$> ls1 []	0	
Λ Λ	\$> >>hola \$> cat hola		ОК		\$> \$>	0	
^ ^	\$> echo hola > bonjour		ОК	ОК	hola	0	
^ ^	\$> cat < bonjour \$> echo hola >bonjour		ОК		\$> hola	0	
^	<pre>\$> cat <bonjour \$=""> echo hola>bonjour</bonjour></pre>		ок		\$> hola	0	
^	<pre>\$> cat<bonjour \$=""> echo hola> bonjour</bonjour></pre>			OK .	\$> hola	0 0	
Ž	<pre>\$> cat< bonjour</pre>		ОК		\$>	0	
^ ^	<pre>\$> echo hola</pre>		ОК	OK .	hola \$>	0	
^	<pre>\$> echo hola</pre>		ОК	O K	hola \$>	0	
Λ Λ	<pre>\$> echo hola > srcs/bonjour \$> cat < srcs/bonjour</pre>		ОК	ок	hola \$>	0	
^	<pre>\$> echo hola >srcs/bonjour \$> cat <srcs bonjour<="" pre=""></srcs></pre>		ок	ОК	hola \$>	0 0	
NS v	<pre>\$> echo hola > bonjour \$> echo que tal >> bonjour</pre>		ОК	ОК	hola que tal	0	
<u> </u>	<pre>\$> cat < bonjour \$> echo hola > bonjour</pre>				<pre>\$> que tal \$></pre>	0	
ECT	<pre>\$> rm bonjour \$> echo que tal >> bonjour \$> cat < bonjour</pre>		ОК	ОК	\$>	0 0 0	
EDIR	<pre>\$> e'c'"h"o hola > bonjour \$> cat 'bo'"n"jour</pre>		ок	ок	hola \$>	0	
R A	\$> echo hola > bonjour\ 1	1011 0555			bonjour 1' []	0	
¥	\$> cat bonjour\ 1	ION GERE	ОК	ОК	<pre>\$> hola \$></pre>	0	

<pre>\$> echo hola > bonjour hey \$> ls</pre>			bonjour [] 0 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
\$> cat bonjour	ОК	ОК	hola hey
\$> cat hey			<pre>cat: hey: No such file or directory</pre>
			\$>
\$> echo hola > srcs/bonjour			bash: prout: No such file or directory
<pre>\$> echo hey > srcs/hello \$> >srcs/bonjour >srcs/hello <pre>cat srcs/bonjour srcs/hello</pre></pre>	ОК	OK	\$> \$>
\$> cat srcs/bonjour srcs/hello			
\$> echo hola > srcs/boniour			bash: prout: No such file or directory 0
\$> echo hey > srcs/hello			\$> bonjour hello
<pre>\$> rm srcs/bonjour srcs/hello \$> >srcs/bonjour >srcs/hello <pre>prout</pre></pre>	ОК	OK	\$> 1
\$> ls srcs			\$>
<pre>\$> cat srcs/bonjour srcs/hello \$> echo hola > srcs/bonjour</pre>			bash: prout: No such file or directory 0
\$> echo hev > srcs/hello			\$>
<pre>\$> >srcs/bonjour <pre>prout >srcs/hello</pre></pre>	OK	OK	\$>
<pre>\$> cat srcs/bonjour \$> cat srcs/hello</pre>			hey \$> 0
			bash: prout: No such file or directory 0
<pre>\$> echo hola > srcs/bonjour \$> echo hey > srcs/hello</pre>			<mark></mark>
<pre>\$> rm srcs/bonjour srcs/hello \$> >srcs/bonjour <pre><pre>prout >srcs/hello</pre></pre></pre>	OK	OK	bonjour 0 1
\$> ls srcs			\$>
\$> cat srcs/bonjour			0
<pre>\$> echo hola >/bonjour \$> echo hey >/hello</pre>			bash: prout: No such file or directory
\$> >/bonjour >/hello <prout \$=""> cat/bonjour/hello</prout>	ОК	OK	\$ \$>
			0
<pre>\$> echo hola >/bonjour \$> echo hey >/hello</pre>			bash: prout: No such file or directory
\$> ecno ney >/netto \$> rm/bonjour/hello	OK	01/	bonjour hello
<pre>\$> rm/bonjour/hello \$> >/bonjour >/hello <prout< pre=""></prout<></pre>	ОК	ОК	\$>
<pre>\$> ls \$> cat/bonjour/hello</pre>			
\$> echo hola >/bonjour			bash: prout: No such file or directory 0
\$> echo hev >/hello			<mark></mark>
<pre>\$> >/bonjour <pre></pre></pre>	ОК	OK	\$>
\$> cat/bonjour \$> cat/hello			\$>
\$> echo hola >/bonjour			bash: prout: No such file or directory 0
<pre>\$> echo hey >/hello \$> rm/bonjour/hello</pre>			\$> 0
\$> >/bonjour <pre><pre><pre></pre></pre></pre>	ОК	OK	bonjour 9 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
<pre>\$> ls \$> cat/bonjour</pre>			\$>
\$> echo hola > srcs/bonjour			bash: prout: No such file or directory 0
\$> echo hota > srcs/hello \$> echo hey > srcs/hello \$> >srcs/bonjour >>srcs/hello <prout< td=""><td></td><td></td><td></td></prout<>			
<pre>\$> >srcs/bonjour >>srcs/hello <prout \$=""> cat srcs/bonjour</prout></pre>	ОК	OK	\$> box
\$> cat srcs/bonjour \$> cat srcs/hello			hey
\$> echo hola > srcs/bonjour			bash: prout: No such file or directory 0
<pre>\$> echo hey > srcs/hello \$> >>srcs/bonjour >srcs/hello <pre>prout</pre></pre>	ок	ОК	\$> bonjour 1
\$> cat srcs/bonjour	OK	OK .	\$>
<pre>\$> cat srcs/hello</pre>			\$>
\$> echo hola > srcs/bonjour			bash: prout: No such file or directory
\$> echo hey > srcs/hello \$> >>srcs/bonjour >>srcs/hello <prout< td=""><td>ок</td><td>ОК</td><td>bonjour 1</td></prout<>	ок	ОК	bonjour 1
5> cat srcs/bonlour	UK	OK	
\$> cat srcs/hello			hey \$>
\$> echo hola > srcs/bonjour			bash: prout: No such file or directory 0
<pre>\$> echo hola > srcs/bonjour \$> echo hey > srcs/hello</pre>	0.14	616	<mark>0</mark> \$>
<pre>\$> >srcs/bonjour <pre>rout >>srcs/hello</pre></pre>	ОК	OK	\$>
\$> cat srcs/bonjour \$> cat srcs/hello			hey \$> 0
<pre>\$> echo hola > srcs/bonjour \$> echo hey > srcs/hello</pre>			bash: prout: No such file or directory 0
\$> >>srcs/boniour <prout>srcs/hello</prout>			\$> bonjour 1
\$> cat srcs/bonjour \$> cat srcs/hello	ОК	ОК	<mark>0</mark> \$>
\$> cat srcs/hello			hey \$>
			\$>

<pre>\$> echo hola > srcs/bonjour \$> echo hey > srcs/hello \$> >>srcs/bonjour <pre> <pre></pre></pre></pre>			ОК	ОК	<pre>bash: prout: No such file or directory \$> bonjour \$> hey \$></pre>	0 0 1 0 0	
<pre>\$> echo hola > srcs/bonjour \$> echo hey > srcs/hello \$> <prout>>srcs/bonjour >>srcs/hello \$> cat srcs/bonjour \$> cat srcs/hello</prout></pre>			ОК		<pre>bash: prout: No such file or directory \$> bonjour \$> hey \$></pre>	0 0 1 0 0	
<pre>\$> echo hola > bonjour \$> echo hey > hello \$> <bonjour>hello \$> cat bonjour \$> cat hello</bonjour></pre>			ОК	ОК	hola \$> >>	0 0 0 0 0	
<pre>\$> echo hola > bonjour \$> echo hey > hello \$> >bonjour >hello < prout \$> cat bonjour \$> cat hello</pre>			ОК	ОК	<pre>bash: prout: No such file or directory \$> \$> \$> \$></pre>	0 0 1 0 0	
<pre>\$> echo hola > bonjour \$> echo hey > hello \$> rm bonjour hello \$> >bonjour >hello < prout \$> cat bonjour \$> cat hello</pre>			ОК	ОК	<pre>bash: prout: No such file or directory \$> \$> \$> \$></pre>	0 0 0 1 0 0	
<pre>\$> echo hola > bonjour \$> echo hey > hello \$> >bonjour <pre>\$> cat bonjour \$> cat hello</pre></pre>			ОК	ОК	<pre>bash: prout: No such file or directory \$> \$> hey \$></pre>	0 0 1 0 0	
<pre>\$> echo hola > bonjour \$> echo hey > hello \$> rm bonjour hello \$> >bonjour <prout \$="" hello=""> cat bonjour</prout></pre>			ОК	ОК	bash: prout: No such file or directory \$> \$>	0 0 0 1 0	Le fichier hello n'existe pas
<pre>\$> echo hola > bonjour \$> <bonjour cat="" wc="" =""> bonjour \$> cat bonjour</bonjour></pre>			ОК	ОК	0 0 0 \$>	0 0 0	
<pre>\$> rm -f bonjour \$> rm bonjour > bonjour \$> ls -l bonjour</pre>			ОК	ОК	ls: cannot access 'bonjour': No such file or directory	0 0 2	
<pre>\$> export HOLA="bonjour hello" \$> >\$HOLA \$> ls</pre>			ОК	ок	'bonjour hello' [] \$>	0 0	
<pre>\$> export HOLA="bonjour hello" \$> >"\$HOLA" \$> ls</pre>			ок	ОК	'bonjour hello' [] \$>	0 0	
\$> export HOLA="bonjour hello" \$> >\$"HOLA" \$> ls			ОК	ОК	HOLA [] \$>	0 0 0	
<pre>\$> export HOLA="bonjour hello" \$> >\$HOLA>hey \$> ls</pre>			ОК	ОК	'bonjour hello' hey [] \$>	0 0 0	
<pre>\$> export HOLA="bonjour hello" \$> >hey>\$HOLA \$> ls</pre>			ОК	ОК	'bonjour hello' hey [] \$>	0 0 0	
<pre>\$> export HOLA="bonjour hello" \$> >hey>\$HOLA>hey>hey \$> ls</pre>			ок	ОК	'bonjour hello' hey [] \$>	0 0 0	
<pre>\$> export A=hey \$> export A B=Hola D E C="Que Tal" \$> echo \$PROUT\$B\$C > /tmp/a > /tmp/b > /tmp/c \$> cat /tmp/a \$> cat /tmp/b \$> cat /tmp/c</pre>			ОК		\$> \$> \$> HolaQue Tal \$>	0 0 0 0 0	
!!!!! Co	ontenu du fichier ontenu du fichier ontenu du fichier	b: "0123456		orrible"			
\$> <a <b="" <c<="" cat="" td=""><td></td><td></td><td>OK</td><td>ОК</td><td>Prout \$></td><td>0</td><td></td>			OK	ОК	Prout \$>	0	

\$> <a <b="" <c<br="" cat="">\$> cat a \$> cat b \$> cat c	OK	ОК	\$>
<pre>\$> >a ls >b >>c >d \$> cat a \$> cat b \$> cat c \$> cat d</pre>	OK	ОК	\$>
<pre>\$> >a ls >b >>c >d \$> cat a \$> cat b \$> cat c \$> cat d</pre>	OK	ОК	\$>
<pre>\$> echo hola > a > b > c \$> cat a \$> cat b \$> cat c</pre>	ОК		\$>
<pre>\$> mkdir dir \$> ls -la > dir/bonjour \$> cat dir/bonjour</pre>	ОК		total 16 drwx 2 vietdu91 [] . drwxrwxewt 51 root [] [] \$>
\$> <a \$> cat a</a 	OK	OK	Amour Tu es Horrible \$>
<pre>\$> >d cat <a>>e \$> cat a \$> cat d \$> cat e</pre>	OK	ОК	\$> Amour \$> Amour \$> Amour \$> Amour \$> Amour \$>
\$> < a > b cat > hey >> d \$> cat d \$> ls	ОК	ОК	Amour Tu Es Horrible 0 \$> 0 [] a b hey d 0
\$> cat << hola	ОК	ОК	> \$HOME 0 > hola /home/vietdu91 \$>
\$> cat << 'hola'	OK	OK	> \$HOME > hola \$HOME \$>
\$> cat << "hola"	ОК	ОК	> \$HOME
<pre>\$> cat << ho"la"</pre>	ОК	OK	SHOME > hola \$HOME \$>
\$> cat << \$HOME	OK	ОК	> prout
<pre>\$> cat << hola > bonjour \$> cat bonjour</pre>	OK	ОК	<pre>> prout > croutte > antoine pue > hola \$> prout croutte antoine pue</pre> 0 6 6 6 7 8 9 9 9 9 9 9 9 9 9 9 9 9

S
\$> < <hola< td=""></hola<>
OK OK > hola
\$> cat << br/>bash: syntax error near unexpected token `newline' 2 \$>
\$> cat << prout << lol << koala OK OK OK OK OK OK OK OK OK O
\$> prout << lol << cat << koala
\$> << \$hola
\$> << \$"hola"\$"b"
\$> << \$"\$hola"\$\$"b" OK OK
\$> << ho\$la\$"\$a"\$\$"b"
\$> echo hola <<< bonjour OK OK hola \$>
\$> echo hola <<<< bonjour OK OK bash: syntax error near unexpected token `<' \$> 2
\$> echo hola <<<< bonjour OK OK OK bash: syntax error near unexpected token `<<' 2 \$>
\$> ls * NON GERE OK OK OK OK OK OK OK OK OK O
Something the state of the stat
S> NONGERE OK OK Crashtest.c test.c 0 [BONUS] \$> ls *.hola NONGERE OK OK S> \$> cat M*le NONGERE OK OK [affiche le Makefile] 0 \$> [BONUS]
\$> cat M*le NON GERE OK OK [affiche le Makefile] 0 [BONUS]
\$> cat M*ee NON GERE OK OK cat: 'M*ee': No such file or directory 1 [BONUS]
\$> cat Make*file NON GERE OK OK [affiche le Makefile] 0 [BONUS]