

## 

## [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 Comportement attendu Sortie d'exit (echo \$?) test.c Notes ККО ок ко > \n (touche entrée) ОК Exit de l'ancienne commande > [que des espaces] ОК > [que des tabulations] ОК Crl + V + Tab OK Marche aussi pour "#" ОК ОК bash: syntax error near unexpected token `newline' ОК > < >> >> ОК ОК 5> << ОК > <> >>>>> bash: syntax error near ОК unexpected token `>>' bash: syntax error near >>>>>>>>> ОК unexpected token `>>' bash: syntax error near · <<<< ОК unexpected token `<<'

OK OK OK   Sequence of class "s"   Sequence of class								
OK		\$> <<<<<<<<		ок	ОК	<pre>bash: syntax error near unexpected token `&lt;&lt;' \$&gt;</pre>	2	
Compared to the control of the con		\$> > > >		ОК	OK	unexpected token `>'	2	Marche aussi pour <
		\$> >> >> >>		ОК	ОК	bash: syntax error near unexpected token `>>'	2	Marche aussi pour <
		\$> >>> >> >>		ОК	ОК	bash: syntax error near unexpected token `>>'	2	Marche aussi pour <
Solid   Color   Colo		\$> /		ОК	ОК	bash: /: Is a directory \$>	126	
		\$> //		ОК	ОК	bash: //: Is a directory \$>	126	
Section   Sect		\$> /.		ОК	ОК	bash: /.: Is a directory \$>	126	
S   NON GERE		\$> /.///		ОК	OK	\$>	126	
Second   S		\$>		ОК	ОК	bash: //////: Is a directory \$>	126	
STANSMITTED		\$> \\	NON GERE	ОК	ОК	bash: \: command not found	127	
		\$> \\\\	NON GERE	ОК	ОК	bash: \\: command not found	127	
Page			NON GERE	ОК	OK	bash: \\\: command not found	127	
OK								Marche aussi pour "_@°]=+£\$?"
OK OK   Section   Company   Compan		\$>		ок	ОК	unexpected token ` '	2	Marche aussi pour les metacharacters "&();"
OK OK OK   OK   OK   OK   OK   OK   O		\$>   hola		ОК	OK	unexpected token ` '	2	
OK		\$>		ОК	ОК	unexpected token ` '	2	
OK OK   Dash: syntax error near unexpected token '  '	NTAXE)	\$>		ОК			2	
OK OK   Dash: syntax error near unexpected token '  '	VOLEE (SY	\$>		ОК	ОК	unexpected token `  '	2	
\$> &&  OK OK oK bash: syntax error near unexpected token '&&' \$>  \$> &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&	A LA	\$>		ОК	OK	unexpected token `  '	2	
OK OK	CARACT	\$> >> ><		ОК	ОК	unexpected token ` '	2	
OK OK Unexpected token `&&' \$> &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&		\$> &&		ОК	OK	unexpected token `&&'	2	
S> ;;  NON GERE  OK  OK  Unexpected token `&&'  S>  bash: syntax error near  unexpected token `;;'  OK  OK  OK  OK  Unexpected token `;;'		\$> &&&&&		ОК	OK	unexpected token `&&'	2	
NON GERE OK OK unexpected token `;;'		\$> &&&&&&&&&&&&&&		ОК	OK	unexpected token `&&'	2	
\$>		\$> ;;	NON GERE	ОК	OK		2	

\$> ;;;;;	NON GERE	ОК	ОК	bash: syntax error near unexpected token `;;'	2	
\$> ;;;;;;;;;;	NON GERE	ОК	ОК	bash: syntax error near unexpected token `;;'	2	
\$> ()	NON GERE	ОК		<pre>bash: syntax error near unexpected token `)'</pre>	2	
\$> ( ( ) )	NON GERE	ОК		<pre>\$&gt; bash: syntax error near unexpected token `)'</pre>	2	
\$> ( ( ( ( ) ) ) )	NON GERE	ОК		<pre>\$&gt; bash: syntax error near unexpected token `)'</pre>	2	
A . III	NON GERE	ОК	OK	<pre>\$&gt; bash: : command not found \$&gt;</pre>	127	
\$> "hola"		ОК	OK	bash: hola: command not found \$>	127	
\$> 'hola'		ОК	OK	bash: hola: command not found \$>	127	
\$> ''	NON GERE	ОК	OK	bash: : command not found \$>	127	
\$> *	NON GERE	ОК	OK	bash: crashtest.c: command not found \$>	127	[BONUS]
\$> */*	NON GERE	ОК	OK	bash: Docs/attest.cation: command not found \$>	127	[BONUS]
\$> */*	NON GERE	ОК		bash: */*: No such file or directory	127	[BONUS] S'il n'y aucun dossier à l'intérieur
\$> .	NON GERE	ОК	UK	bash: .: filename argument required \$>	2	
\$>	NON GERE	ОК	UK	bash:: command not found \$>	127	
\$> ~	NON GERE	ОК	UK	<pre>bash: /home/vietdu91: Is a directory \$&gt;</pre>	126	
\$> ABC=hola	NON GERE	ОК		\$>	0	/!\ A gerer comme une erreur
\$> 4ABC=hola	NON GERE	ОК	UK	bash: 4ABC=hola: command not found \$>	127	
hola		ОК	UK	bash: hola: command not found \$>	127	
\$> hola que tal		ОК	UK	bash: hola: command not found \$>	127	
\$> Makefile		ОК	UK	bash: Makefile: command not found \$>	127	
\$> echo		ок	OK	\$>	Θ	
\$> echo -n		ОК	ОК	\$>	0	
\$> echo Hola		ОК		Hola \$>	0	
\$> echoHola		ОК	UK	bash: echoHola: command not found \$>	127	
\$> echo-nHola		ОК	UK	bash: echo-nHola: command not found \$>	127	
\$> echo -n Hola		ОК		Hola\$>	0	
\$> echo "-n" Hola		ОК		Hola\$>	0	
\$> echo -nHola		ОК	OK	-nHola \$>	0	
\$> echo Hola -n		ОК	OK	Hola -n \$>	0	
<pre>\$&gt; echo Hola Que Tal</pre>		ОК		Hola Que Tal \$>	0	

\$> echo Hola	ОК	ОК	Hola \$>	0	
\$> echo Hola Que Tal	ОК	ОК	Hola Que Tal \$>	0	
\$> echo \n hola	ОК	ОК	n hola \$>	0	A gerer ?
\$> echo "	ОК	ОК	\$ \$>	0	
\$> echo   cat -e	ОК	ОК	\$ \$>	Θ	
\$> ""''echo hola""''' que""'' tal""''	ОК	OK .	hola que tal \$>	0	
\$> echo -n -n	ОК		\$>	0	
\$> echo -n -n Hola Que	ОК	ОК	Hola Que\$>	0	
\$> echo -p	ОК		-p \$>	0	
\$> echo -nnnnn	OK		\$>	0	
\$> echo -n -nnn -nnnn	ОК	ОК	\$>	0	
\$> echo -n-nnn -nnnn	ОК	ОК	-n-nnn -nnnn \$>	0	
\$> echo -n -nnn hola -nnnn	OK		hola -nnnn\$>	0	
\$> echo -n -nnn-nnnn	OK	ОК	-nnn-nnnn\$>	0	
\$> echon	ОК	ОК	n \$>	0	
\$> echo -nnnn	ОК	ОК	n\$>	0	
\$> echo -nnnnnnn	OK	OK	nnnnnn\$>	0	
\$> echo -nnnnnnn	ОК	ОК	nnnn\$>	0	
\$> echo \$	ОК	ОК	\$ \$>	0	
\$> 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 \$?	OK	ОК	\$> \$:\$=\$	Θ	
<pre>\$&gt; echo \$:\$=   cat -e  \$&gt; echo " \$ "   cat -e</pre>	ОК	OK	\$>	0	
\$> echo ' \$ '   cat -e	ОК		\$ \$ \$>	0	
\$> echo \$HOME	ОК		<pre>\$ \$ \$&gt; /home/vietdu91</pre>	0	
\$> echo \\$HOME	OK	ОК	\$> \$HOME	0	
\$> echo my shit terminal is [\$TERM]	OK	OK .	<pre>\$&gt; my shit terminal is [xterm-256color]</pre>	0	
\$> echo my shit terminal is [\$TERM4	ок ок	OK .	\$> my shit terminal is [	0	
\$> echo my shit terminal is [\$TERM4]	OK	OK .	<pre>\$&gt; my shit terminal is []</pre>	Θ	
\$> echo \$UID	ОК	ОК	\$> 1000	0	
\$> echo \$HOME9	ОК	ОК	\$>		Marche pour tous les caractères alphanumériques
\$> echo \$9HOME	ОК		\$> HOME \$>	Θ	de l'ASCII
\$> echo \$HOME%			/home/vietdu91%	0	Marche pour les autres caractères (ponctuation,
32 ECHO SHOPLA	ОК		\$>		caractères non-ASCII, etc.)

	<pre>\$&gt; echo \$UID\$HOME</pre>		ОК	ОК	1000/home/vietdu91% \$>	0	
	\$> echo Le path de mon HOME est \$HOME		ОК	ОК	Le path de mon HOME est /home/vietdu91 \$>	0	
	<pre>\$&gt; echo \$USER\$var\\$USER\$USER\\$USERtest\$USER</pre>		ОК	ОК	vietdu91\$USERvietdu91\$USERtestvietdu91 \$>		A ne pas gerer si '\' non gere : Doit sortir : vietdu91vietdu91vietdu91
	\$> echo \$hola*		ОК	ОК	\$>	0	Cas d'un variable inexistant
	\$> echo -nnnn \$hola		ОК	ОК	\$>	Θ	
	\$> echo > <		ОК	UK	bash: syntax error near unexpected token `<' \$>	2	
	\$> echo		ОК	UK	bash: syntax error near unexpected token ` ' \$>	2	
	\$> EechoE		ОК	UK	<pre>bash: EechoE: command not found \$&gt;</pre>	127	
	\$> .echo.		ОК	UK	bash: .echo.: command not found \$>	127	
	\$> >echo>		ОК		bash: syntax error near unexpected token `newline' \$>	2	
	\$> <echo<< td=""><td></td><td>ОК</td><td>UK</td><td>bash: syntax error near unexpected token `newline' \$&gt;</td><td>2</td><td></td></echo<<>		ОК	UK	bash: syntax error near unexpected token `newline' \$>	2	
	\$> >>echo>>		ОК	UK	bash: syntax error near unexpected token `newline' \$>	2	
	\$>  echo		ОК	UK	bash: syntax error near unexpected token ` ' \$>	2	
	\$>  echo -n hola		ОК	UK	bash: syntax error near unexpected token ` ' \$>	2	
	\$> echo *	NON GERE	ОК	ок	<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$&gt;</pre>	0	[BONUS]
2	\$> echo '*'	NON GERE	ОК	ОК	* \$>	0	Marche aussi pour les double quotes
CHO	\$> echo D*	NON GERE	ОК	ОК	Docs Dockers Drawings \$>	0	[BONUS]
ш	\$> echo *Z	NON GERE	ОК	ОК	*Z \$>	0	[BONUS]
	\$> echo *t hola	NON GERE	ОК	()K	<pre>crashtest.c test.c hola \$&gt;</pre>	0	[BONUS]
	\$> echo *t	NON GERE	ОК	UK	<pre>crashtest.c test.c \$&gt;</pre>	0	[BONUS]
	\$> echo \$*	NON GERE	ОК	ок	\$>	0	[BONUS]
	<pre>\$&gt; echo hola*hola *</pre>	NON GERE	ОК	ОК	hola*hola crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$>	0	[BONUS]
	\$> echo \$hola*	NON GERE	ОК		<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$&gt;</pre>	0	[BONUS]
	\$> echo \$HOME*	NON GERE	ОК	ок	/home/vietdu91 \$>	0	[BONUS]
	\$> echo \$""		ОК	ОК	\$>	0	
	\$> echo "\$"""		ОК	ОК	\$ \$>	0	
	\$> echo '\$'''		ОК	ОК	\$ \$>	0	
	\$> echo \$"HOME"		ОК		HOME \$>	0	
	\$> echo \$''HOME		ОК		HOME \$>	0	

Company	\$> echo \$""HOME	01/	211	HOME	0	
		ОК	OK	\$>		
	\$> echo "\$H0"ME	ОК			θ	
	\$> echo '\$H0'ME	ок			0	
Seed	\$> echo "\$H0""ME"	ок			0	
Seed of Section   Seed of Section   Seed of Section   Seed of Section   Se	\$> echo '\$H0''ME'	ОК	OK	\$HOME	Θ	
	\$> echo "'\$H0''ME'"	ок		''ME'		ne sais pas pourquoi Excel tronque à
Section   Sect	\$> echo ""\$HOME	ОК			0	
Section   Sect	\$> echo "" \$HOME	ок	OK	/home/vietdu91		
	\$> echo ''\$HOME	ок	OK	/home/vietdu91	0	
	\$> echo '' \$HOME	ОК	OK	/home/vietdu91		
Seed of Shore   Shor	\$> echo \$"HO""ME"	ОК	OK	НОМЕ	О	
See etho \$1000000000000000000000000000000000000	\$> echo \$'HO''ME'	ОК	OK	НОМЕ	О	
Section SHORE	\$> echo \$'HOME'	ок	OK	НОМЕ	О	
See etho S-HORE	\$> echo "\$"HOME	ОК	OK	\$HOME	Θ	
See echo \$'HOLA'	\$> echo \$=HOME	ОК			0	
Some	\$> echo \$"HOLA"	ОК			0	
Section   Sport Exist   Hola   Section   Hola   H	\$> echo \$'HOLA'	ОК			0	
S   S   S   S   S   S   S   S   S   S	\$> echo \$DONTEXIST Hola	ОК	OK	Hola		
S   OK   OK   S   S   S   S   S   S   S   S   S	\$> echo "hola"	ОК	ОК	hola \$>		
S   Cho ''hola''	\$> echo 'hola'	ок	OK	hola	Θ	
\$> echo ''h'o'la''         0K         OK         OK         hola s>         0<	\$> echo ''hola''	ОК	OK	hola	0	
\$> echo "''h'o'la''" \$> echo "''h'o'la''" \$> echo "''h'o'la''' \$> echo "'hola'"  \$> echo "'hola"'  \$> echo "hola"'  \$> echo "hola"  \$> echo "hola	\$> echo ''h'o'la''	ОК	OK	hola	0	
Secho ""h'o'la""	\$> echo "''h'o'la''"	ОК	OK	'h'o'la''	0	
\$> echo"'hola'"  \$> echo "'hola'"  OK OK OK  OK OK  'hola'  \$> echo '"hola"'  OK OK  OK  OK  OK  OK  OK  OK  OK  OK	\$> echo "'"h'o'la"'"	ОК	OK	hola'	0	
\$> echo "'hola'"  \$> echo '"hola"'  OK OK OK  OK OK  *hola"  \$> echo '"hola"'  OK OK  OK  OK  OK  OK  OK  OK  OK  OK	\$> echo"'hola'"	ОК	OK	bash: echo'hola': command not found	0	
\$> echo '"hola"'	\$> echo "'hola'"	ОК	OK	'hola'	0	
\$> echo '''ho"'''l"a'''	\$> echo '"hola"'	ОК	OK	"hola"	0	
	\$> echo '''ho"'''l"a'''	ОК	OK	ho"l"a	0	
OR OR \$>	\$> echo hola""""""""""""""""""""""""""""""""""""	ОК		hola \$>	0	

	\$> echo hola"''''	ОК	ОК	hola'''''' \$>	0	
	\$> echo hola'''''	ОК	OK .	hola \$>	0	
	\$> echo hola'"""""""""	ОК	ОК	hola""""""""""""""""""""""""""""""""""""	0	
	\$> e"cho hola"	ОК	ОК	bash: echo hola: command not found \$>	127	
	\$> e'cho hola'	ОК	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		hola \$>	0	
	\$> echo ""hola	ок	ОК	hola \$>	Θ	
	\$> echo "" hola	ОК	ОК	hola \$>	Θ	
	\$> 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	ОК	bash: e'c'ho 'b'onjo'u'r: command not found \$>	127	
	\$> echo "\$DONTEXIST"Makefile	ок	ОК	Makefile \$>	Θ	
	\$> echo "\$DONTEXIST""Makefile"	ок	ОК	Makefile \$>	Θ	
	\$> echo "\$DONTEXIST" "Makefile"	ок		Makefile \$>	Θ	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	OK .	bash: ?/home/vietdu91: command not found \$>	127	
	\$> \$	ОК	OK .	bash: \$: command not found \$>	127	
<b>₩</b>	\$> \$HOME	ок	OK .	<pre>bash: /home/vietdu91: Is a directory \$&gt;</pre>	126	
	\$> \$HOMEdskjhfkdshfsd	ОК	OK	\$>	0	
	\$> "\$HOMEdskjhfkdshfsd"	ОК	ОК	bash: : command not found \$>	127	
	\$> '\$HOMEdskjhfkdshfsd'	ОК	ОК	<pre>bash: \$HOMEsddfddfssfdfs: command not found \$&gt;</pre>	127	
	\$> \$DONTEXIST	ОК	ОК	\$>	0	Cas d'un variable inexistant
	\$> \$LESS\$VAR	ОК		bash: -R: command not found \$>	127	
	\$> Ctlr-C	ОК	ОК	\$> ^C \$>	130	
	\$> holaCtlr-C	ОК	ок	<pre>\$&gt; hola^C \$&gt;</pre>	130	

	<pre>\$&gt; cat (faire Ctlr-C apres avoir fait plusieurs fois [ENTREE])</pre>		ОК	ОК	^C	130	
	<pre>\$&gt; cat   rev(faire Ctlr-C apres avoir fait plusieurs fois [ENTREE])</pre>		ОК	ОК	^C	0	
SIGNAU	\$> Ctlr-D		ОК	ОК	<pre>\$&gt; \$&gt; exit ~/minishell \$&gt;</pre>	0	Sort du programme
	\$> hola Ctlr-D		ОК	ОК	\$> hola	/	Ne fait rien
	\$> Ctlr-\		ОК	ОК	\$>	•	Ne fait rien
	\$> hola Ctlr-\		ОК	ОК	\$> hola	/	
	<pre>\$&gt; cat (faire Ctlr-\ apres avoir fait plusieurs fois [ENTREE])</pre>		ОК		^\Quit (core dumped) \$>	131	
	\$> Ctlr-Z		ОК	ОК	\$>	/	Ne fait rien
	<pre>\$&gt; sleep 3   sleep 3   sleep 3 (faire Ctlr-C une seconde apres)</pre>		ОК	( ) K	^C \$>	130	
	<pre>\$&gt; sleep 3   sleep 3   sleep 3 (faire Ctlr-D une seconde apres)</pre>		ок	ОК	\$>	Θ	Ne fait rien
	<pre>\$&gt; sleep 3   sleep 3   sleep 3 (faire Ctlr-\ une seconde apres)</pre>		ОК	UK	^\Quit (core dumped) \$>	131	
	\$> env		ок	ОК	 PATH= \$>	0	
	\$> env hola	NON GERE	ОК	UK	<pre>env: 'hola': No such file or directory \$&gt;</pre>	127	
	\$> env hola que tal	NON GERE	ок	UK	env: 'hola': No such file or directory \$>	127	
	\$> env env	NON GERE	ОК	ОК	 PATH= \$>	Θ	Affiche une seule fois env
	\$> env env env env	NON GERE	ОК	OK	PATH= \$>	0	Affiche une seule fois env
	\$> env ls	NON GERE	ок	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$&gt;</pre>	0	
	\$> env ./Makefile	NON GERE	ОК	UK	env: './Makefile': Permission denied \$>	126	
	<pre>\$&gt; export HOLA=bonjour \$&gt; env</pre>		ОК	ОК	HOLA=bonjour \$>	0 0	
	<pre>\$&gt; export HOLA=bonjour \$&gt; env</pre>		ок		HOLA=bonjour \$>	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>\$&gt; export Hola \$&gt; export</pre>		ОК	ОК	export HOME="/home/vietdu91" export Hola export PATH="" \$>	0	Hola n'existe pas dans env mais il existe dans export

<pre>\$&gt; export Hola9hey \$&gt; export</pre>		ОК	ОК	export HOME="/home/vietdu91" export Hola9hey export PATH="" \$>	0 0	
\$> export \$DONTEXIST		ОК	ОК	export HOME="/home/vietdu91" export PATH="" \$>	θ	Le cas ou la variable n'existe pas et n'est pas importee
\$> export   grep "HOME"		ок	ОК	export HOME="/home/vietdu91" \$>	0	
\$> export ""		ОК	ок	bash: export: `': not a valid identifier \$>	1	
\$> export =		ОК	ОК	bash: export: '=': not a valid identifier \$>	1	
\$> export %		ОК	ОК	bash: export: '%': not a valid identifier \$>	1	
<pre>\$&gt; export \$?</pre>		ОК	ОК	bash: export: `0': not a valid identifier \$>	1	
\$> export ?=2		ОК	ОК	bash: export: '?=2': not a valid identifier	1	
\$> export 9HOLA=		ОК	ОК	bash: export: '9HOLA=': not a valid identifier \$>	1	
<pre>\$&gt; export HOLA9=bonjour \$&gt; env</pre>		ОК	ОК	 HOLA9=bonjour \$>	0	
<pre>\$&gt; export _HOLA=bonjour \$&gt; env</pre>		ОК	ОК	 _HOLA=bonjour \$>	0	
<pre>\$&gt; exportHOLA=bonjour \$&gt; env</pre>		ОК	ОК	 HOLA=bonjour \$>	0 0	
<pre>\$&gt; export _HO_LA_=bonjour \$&gt; env</pre>		ОК	ок	 _HO_LA_=bonjour \$>	0	
\$> export HOL@=bonjour		ОК	ок	<pre>bash: export: 'HOL@=bonjour': not a valid identifier \$&gt;</pre>	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>\$&gt; export -HOLA=bonjour</pre>		ОК	ОК	bash: export: -H: invalid option	2	
<pre>\$&gt; exportHOLA=bonjour</pre>		ОК	ОК	bash: export:: invalid option	2	
<pre>\$&gt; export HOLA-=bonjour</pre>		ок	ок	<pre>bash: export: `HOLA-=bonjour': not a valid identifier \$&gt;</pre>	1	
\$> export HO-LA=bonjour		ОК	ОК	bash: export: `HO-LA=bonjour': not a valid identifier \$>	1	
\$> export HOL.A=bonjour		ок	ОК	bash: export: `HOL.A=bonjour': not a valid identifier \$>	1	
\$> export HOL\\\\$A=bonjour	NON GERE	ок	ок	<pre>bash: export: `HOL\\$A=bonjour': not a valid identifier \$&gt;</pre>	1	
\$> export H0\\\\LA=bonjour	NON GERE	ОК	ОК	bash: export: `HO\\LA=bonjour': not a valid identifier \$>	1	
\$> export HOL}A=bonjour		ОК	ОК	bash: export: `HOL}A=bonjour': not a valid identifier \$>	1	
<pre>\$&gt; export HOL{A=bonjour</pre>		ОК	ОК	bash: export: `HOL{A=bonjour': not a valid identifier \$>	1	
\$> export H0*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: export: `HO@LA=bonjour': not a valid identifier \$>	1	
				4		

	\$> export HO!LA=bonjour		ОК	ОК	bash: !LA=bonjour: event not found	0	
	<pre>\$&gt; export HO\$?LA=bonjour \$&gt; env</pre>		ОК	ок	HOOLA=bonjour \$>	0	
	\$> export +HOLA=bonjour		ок	ок	bash: export: `+HOLA=bonjour': not a valid identifier \$>	1	
	\$> export HOL+A=bonjour		ОК	ОК	bash: export: `HOL+A=bonjour': not a valid identifier \$>	1	
	<pre>\$&gt; export HOLA+=bonjour \$&gt; env</pre>	NON GERE	ОК	ОК	 HOLA=bonjour \$>	0	
	<pre>\$&gt; export HOLA=bonjour \$&gt; export HOLA+=bonjour \$&gt; env</pre>	NON GERE	ОК	ок	 HOLA=bonjourbonjour \$>	0	"+=" s'ajoute au contenu de la variable
	<pre>\$&gt; exportHOLA=bonjour \$&gt; env</pre>	NON GERE	ОК	ок	PATH= \$>	0	
	\$> export HOLA =bonjour		ок	ок	bash: export: '=bonjour': not a valid identifier \$>	1	
	\$> export HOLA = bonjour		ОК	ОК	bash: export: '=': not a valid identifier \$>	1	
	<pre>\$&gt; export HOLA=bon jour \$&gt; env</pre>		ОК	ок	HOLA=bon \$>	0	
	<pre>\$&gt; export HOLA= bonjour \$&gt; env</pre>		ОК	ок	 HOLA= \$>	0	
	<pre>\$&gt; export HOLA=bonsoir \$&gt; export HOLA=bonretour \$&gt; export HOLA=bonjour \$&gt; env</pre>		ОК	ок	 HOLA=bonjour \$>	0 0 0	
	\$> export HOLA=\$HOME \$> env		ОК	ок	HOLA=/home/vietdu91 \$>	0 0	
	<pre>\$&gt; export HOLA=bonjour\$HOME \$&gt; env</pre>		ОК	ок	HOLA=bonjour/home/vietdu91 \$>	0 0	
	<pre>\$&gt; export HOLA=\$HOMEbonjour \$&gt; env</pre>		ОК		HOLA= \$>	0 0	
	<pre>\$&gt; export HOLA=bon\$jour \$&gt; env</pre>		ОК	ок	HOLA=bon \$>	0 0	Le cas où la variable "\$jour" n'existe pas
	<pre>\$&gt; export HOLA=bon\jour \$&gt; env</pre>	NON GERE	ОК	ок	HOLA=bonjour \$>	0 0	
UNSET	<pre>\$&gt; export HOLA=bon\\jour \$&gt; env</pre>	NON GERE	ок	ок	HOLA=bon\jour \$>	0 0	
∞	\$> export HOLA=bon(jour	NON GERE	ок		bash: syntax error near unexpected token `(' \$>	2	Marche pour les caractères : ()
EXPORT	\$> export HOLA=bon()jour	NON GERE	ок		bash: syntax error near unexpected token `(' \$>	2	
ENV &	\$> export HOLA=bon&jour	NON GERE	ОК	ок	bash: jour: command not found \$>	127	Marche pour les caractères : &
	<pre>\$&gt; export HOLA=bon@jour \$&gt; env</pre>		ок	OK	<pre>\$&gt; env HOLA=bon@jour \$&gt;</pre>	0	

<pre>\$&gt; export HOLA=bon;jour</pre>			bash: jour: command not found	127	
\$> env	ОК	ОК	\$> env 	0	
			HOLA=bon		
\$> export HOLA=bon!jour	ОК	OK	<pre>\$&gt; bash: !jour: event not found</pre>	1	
\$> export HOLA=bon"jour"			\$>	Δ	
\$> export notA=bon jour \$> env	ОК		HOLA=bonjour	0	
			PATH= \$>		
\$> export HOLA\$USER=bonjour				0	
\$> env	ОК		HOLAvietdu91=bonjour PATH=	O	
\$> export HOLA=bonjour=casse-toi			\$> HOLA=bonjour=casse-toi	0	
\$> expoint HOLA=bonijour=casse-tot \$> echo \$HOLA	ОК	ОК	S>	0	
<pre>\$&gt; export "HOLA=bonjour"=casse-toi \$&gt; echo \$HOLA</pre>	ок	ок	HOLA=bonjour=casse-toi \$>	0	
\$> export HOLA=bonjour	01/		bonjour et casse-toi	0	
<pre>\$&gt; export BYE=casse-toi \$&gt; echo \$HOLA et \$BYE</pre>	ОК	ОК	\$>	0	
<pre>\$&gt; export HOLA=bonjour BYE=casse-toi \$&gt; echo \$HOLA et \$BYE</pre>	ОК	ок	bonjour et casse-toi \$>	0	
\$> export \$HOLA=bonjour				0	
\$> env	ок	ОК	bonjour=bonjour PATH=		
			\$>		
<pre>\$&gt; export HOLA="bonjour" \$&gt; echo \$HOLA   cat -e</pre>	ОК	ок	bonjour\$ \$>	θ	
<pre>\$&gt; export HOLA=" -n bonjour " \$&gt; echo \$HOLA</pre>	ок	ок	bonjour\$>	0	
\$> export HOLA="bonjour"/ \$> echo \$HOLA	ОК	ОК	bonjour / \$>	0	
<pre>\$&gt; export HOLA='"' \$&gt; echo " \$HOLA "   cat -e</pre>	ок	ок	" \$ \$>	Θ	
\$> export HOLA=at \$> c\$HOLA Makefile	ок	ОК	[cela affiche le Makefile] \$>	0	
\$> export "" HOLA=bonjour			bash: export: `": not a valid identifier	1	
\$> env	ОК	OK	\$> env 		
			HOLA=bonjour \$>		
<pre>\$&gt; export HOLA="cat Makefile   grep NAME" \$&gt; echo \$HOLA</pre>	ОК	ОК	cat Makefile   grep NAME \$>	0	
\$> export HOLA=hey \$> echo \$HOLA\$HOLA\$HOLA=hey\$HOLA	ок	ОК	heyheyhey=heyhey \$>	0	
\$> export HOLA=" bonjour hey " \$> echo \$HOLA   cat -e	ОК	ОК	bonjour hey\$ \$>	0	
\$> export HOLA=" bonjour hey " \$> echo """\$HOLA"""   cat -e	ОК	ОК	bonjour hey \$ \$>	0	
\$> export HOLA=" bonjour hey " \$> echo wesh"\$HOLA"   cat -e	ОК	ОК	wesh bonjour hey \$ \$>	0	
\$> export HOLA=" bonjour hey " \$> echo wesh""\$HOLA.	ОК	ОК	wesh bonjour hey . \$>	0 0	<pre>/!\ Bien supprimer les espaces en trop au debut et a la fin du string d'export</pre>
\$> export HOLA=" bonjour hey " \$> echo wesh\$""HOLA.	ОК	ОК	weshHOLA. \$>	0	
\$> export HOLA=" bonjour hey " \$> echo wesh\$"HOLA HOLA".	ОК	ОК	weshHOLA HOLA. \$>	0	
\$> export HOLA=bonjour \$> export HOLA=" hola et \$HOLA"	OK	ОК	hola et bonjour \$>	0	
\$> echo \$HOLA				0	

<pre>\$&gt; export HOLA=bonjour \$&gt; export HOLA=' hola et \$HOLA' \$&gt; echo \$HOLA</pre>	ок		hola et \$HOLA
<pre>\$&gt; export HOLA=bonjour \$&gt; export HOLA=" hola et \$HOLA"\$HOLA \$&gt; echo \$HOLA</pre>	ок		hola et bonjourbonjour 0 0 5> 0 0
\$> export HOLA="ls -l - a" \$> echo \$HOLA	ОК	OK	ls -l - a
<pre>\$&gt; export HOLA="s -la" \$&gt; l\$HOLA</pre>			total 16 0 drwx 2 vietdu91 [] . 0
	ОК	OK .	<pre>drwxrwxewt 51 root []rwxr-xr-x 8 vietdu91 [] crashtest.c [] \$&gt;</pre>
<pre>\$&gt; export HOLA="s -la" \$&gt; l"\$HOLA"</pre>	ОК	UK	bash: ls -la: command not found 127 \$>
<pre>\$&gt; export HOLA="s -la" \$&gt; l'\$HOLA'</pre>	ОК	UK	bash: l\$HOLA: command not found 127 \$>
<pre>\$&gt; export HOLA="l" \$&gt; \$HOLAs</pre>	ОК	ОК	\$>
<pre>\$&gt; export HOLA="l" \$&gt; "\$HOLA"s</pre>	ок		<pre>crashtest.c Docs Dockers Drawings Makefile minishell  philosophers srcs test.c \$&gt;</pre>
<pre>\$&gt; export HOL=A=bonjour \$&gt; env</pre>	ок		HOL=A=bonjour \$>
<pre>\$&gt; export HOLA="l" \$&gt; '\$HOLA's</pre>	ОК	UK	bash: \$HOLAs: command not found 127 \$>
<pre>\$&gt; export HOL=A="" \$&gt; env</pre>	ок		0 HOL=A= 0 \$>
<pre>\$&gt; export TE+S=T \$&gt; env</pre>	ОК	UK	<pre>bash: export: `TE+S=T': not a valid identifier \$&gt;</pre>
<pre>\$&gt; export ""=""</pre>	ОК	UK	<pre>bash: export: `=': not a valid identifier 1 \$&gt;</pre>
\$> export ''=''	ОК	UK	bash: export: `=': not a valid identifier 1 \$>
\$> export "="="="	ОК	OK	bash: export: `===': not a valid identifier 1 \$>
\$> export '='='='	ОК	UK	bash: export: `===': not a valid identifier 1
<pre>\$&gt; export HOLA=p \$&gt; export BYE=w \$&gt; \$HOLA"BYE"d</pre>	ок		bash: pBYEd: command not found 127 \$>
<pre>\$&gt; export HOLA=p \$&gt; export BYE=w \$&gt; "\$HOLA"'\$BYE'd</pre>	ОК		<pre>bash: p\$BYEd: command not found \$&gt;</pre>
<pre>\$&gt; export HOLA=p \$&gt; export BYE=w \$&gt; "\$HOLA""\$BYE"d</pre>	ОК	ок	/home/vietdu91/42_works/minishell 0
<pre>\$&gt; export HOLA=p \$&gt; export BYE=w \$&gt; \$"HOLA"\$"BYE"d</pre>	ОК		bash: HOLABYEd: command not found \$>
<pre>\$&gt; export HOLA=p \$&gt; export BYE=w \$&gt; \$'HOLA'\$'BYE'd</pre>	ОК		bash: HOLABYEd: command not found \$>
<pre>\$&gt; export HOLA=-n \$&gt; "echo \$HOLA" hey</pre>	ОК	UK	bash: echo -n: command not found 127 \$>
\$> 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"	ок	UK	1'2\$C"\$D5"\$F'7'8 0 0

<ul> <li>export HOLA=bonjour</li> <li>env</li> <li>unset HOLA</li> <li>env</li> </ul>		ОК	ок	PATH= \$>	0 0 0	
> export HOLA=bonjour					0	
> export flock-bonjour				PATH=	9	
> unset HOLA		ОК		\$>	0	
> unset HOLA					0	
> env > unset PATH					9	
>> unset PATH >> echo \$PATH		ОК	ОК	\$>	0	
> unset PATH		Olí	01/	bash: ls: No such file or directory	127	
> ls		ОК	ОК			
> unset ""		ок	UK	bash: unset: `': not a valid identifier	1	
				\$>		
> unset INEXISTANT		ОК		\$>	0	Cas où la variable n'existe pas
5> unset PWD 5> env   grep PWD				\$> \$>	0	
> pwd		ОК	ОК	/home/vietdu91/42_works/minishell	0	
				\$>		
5> pwd 5> unset PWD				/home/vietdu91/42_works/minishell	0	
>> unset rwb >> env   grep PWD		ОК	ок	\$>	0	
5> cd \$PWD				\$>	0	
> pwd				/home/vietdu91	0	
o> unset OLDPWD o> env   grep OLDPWD		ОК	ОК	\$> \$>	0	
> unset 9HOLA				bash: unset: `9HOLA': not a valid identifier		
disce short		OK	UK	\$>	1	
5> unset HOLA9		ОК	ОК	\$>	0	
> unset HOL?A		ОК	ОК	bash: unset: `HOL?A': not a valid identifier		
		OK		\$>	1	
5> unset HOLA HOL?A		ОК	UK	<pre>bash: unset: `HOL?A': not a valid identifier \$&gt;</pre>	1	Il unset bien HOLA
> unset HOL?A HOLA				bash: unset: `HOL?A': not a valid identifier	1	Il unset bien HOLA
disce lide. A fidea		ОК	ОК	\$>	1	Te disce bell flora
> unset HOL?A HOL.A				bash: unset: `HOL?A': not a valid identifier		
		ОК		bash: unset: `HOL.A': not a valid identifier		
> unset HOLA=				<pre>\$&gt; bash: unset: `HOLA=': not a valid identifier</pre>	1	
> unset nota=		ОК	OK	\$>	1	
> unset HOL\\\A	NON GERE	ОК	ОК	bash: unset: `HOL\\A': not a valid identifier	1	
	NON GERE	ОК		bash: A: command not found	127	
> 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		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 -HOLA		ОК		bash: unset: -H: invalid option	2	
> unset _HOLA		ОК		\$>	0	
> unset HOL_A		ОК	ОК	\$>	0	
> unset HOLA_		ОК	ОК	\$>	0	
> 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 \$HOLA		ОК	OK	\$>	0	Il va unset le contenu de \$HOLA, c'est-a-dire la variable bonjour
s> unset \$PWD		ОК	ОК	<pre>bash: unset: `/home/vietdu91/42_works/minishell': not a valid identifier</pre>	1	

	>> unset HOL@	ОК	ОК	bash: unset: `HOL@': not a valid identifier 1	
	>> unset HOL!A	ОК		bash: !A: event not found 1	
	>> unset HOL^A	ОК		bash: unset: `HOL^A': not a valid identifier 1	
	>> unset HOL\$?A	ОК		\$> 0	Unset prend en compte le contenu de \$?, c'est-a-dire qu'il lit HOLOA ou HOL127A
	>> unset HOL\~A	ОК	ОК	bash: unset: `HOL~': not a valid identifier 1	·
	> unset "" HOLA >> env   grep HOLA	ОК	ОК	bash: unset: `': not a valid identifier 1	HOLA sera quand meme unset !!
	\$> unset PATH \$> echo \$PATH	ОК	ОК	\$> 0	
	> unset PATH > cat Makefile	ОК	UK	bash: cat: No such file or directory \$> 1	
	> unset =	ОК		bash: unset: `=': not a valid identifier 0	
	s> unset =====	ОК		bash: unset: `=====': not a valid identifier 0	
	> unset +++++	ОК		bash: unset: `++++++': not a valid identifier 0	
	> unset	OK		\$> 0	
	> unset export	OK		\$> 0	
	> unset echo	OK		\$> 0	
	> unset pwd	OK		\$> 0	
	> unset cd	OK OK		\$> 0	
	>> unset unset >> unset sudo	OK		\$> 0 \$> 0	
	>> export hola   unset hola   echo \$?			3>	
	>> /bin/echo	ОК		\$> 	
SSO		ОК		\$>	
TIER ES	> /bin/echo Hola Que Tal	ОК	UK	Hola Que Tal 0 \$>	
FICH	>> /bin/env	ОК		0 PATH= \$>	
<u> </u>	5> /bin/cd Desktop	ОК	ОК	bash: /bin/cd: No such file or directory 127	
HISTORIQUE	history  NON GERE	ОК	ОК	558 echo \$? 559 clear 560 history \$>	Affiche les 500 dernières commandes
I	>> [touche du haut]	ОК	ОК	\$> clear /	Defile l'historique
	>> pwd	ОК	ОК	<pre>/home/vietdu91/42_works/minishell \$&gt;</pre>	Retourne \$PWD
	>> pwd hola	ОК	UK	/home/vietdu91/42_works/minishell	
	\$> pwd ./hola	ОК	ОК	/home/vietdu91/42_works/minishell 0 \$>	
	>> pwd hola que tal	ОК		/home/vietdu91/42_works/minishell 0 \$>	
	\$> pwd -p	ОК	UK	bash: pwd: -p: invalid option 2 \$>	
	\$> pwdp	ОК	UK	bash: pwd:: invalid option 2 \$>	
	\$> pwdp	ОК	UK	bash: pwd:: invalid option 2 \$>	
	\$> pwd p	ОК	OK	/home/vietdu91/42_works/minishell 0 \$>	
	\$> pwd pwd pwd	ОК	OK .	/home/vietdu91/42_works/minishell 0 \$>	N'affiche qu'une fois
	\$> pwd ls	ОК		<pre>/home/vietdu91/42_works/minishell \$&gt;</pre>	

	\$> pwd ls env		ОК	ОК	/home/vietdu91/42_works/minishell \$>	0	
	\$> cd		ОК	ок	\$>		On doit se retrouver dans le dossier défini par \$HOME /home/vietdu91
	\$> cd .		ОК	ок	\$>		L'user reste dans le même dossier /home/vietdu91/42_works/minishell
	\$> cd ./		ОК	ОК	\$>		/home/vietdu91/42_works/minishell
	\$> cd ././.		ОК		\$>		/home/vietdu91/42_works/minishell
	\$> cd ./././		ОК		\$>		/home/vietdu91/42_works/minishell
	\$> cd				\$>		On se retrouve au dossier précédent
			OK	ОК			/home/vietdu91/42_works
	\$> cd/		ОК	ОК	\$>	Θ	/home/vietdu91/42_works
	\$> cd/		ОК	OK	\$>	Θ	/home/vietdu91
	\$> cd/.		ОК	ОК	\$>	Θ	/home/vietdu91/42_works
	\$> cd/././.		ОК	ОК	\$>	0	/home/vietdu91/42_works
	\$> cd srcs		ОК	ОК	\$>	0	/home/vietdu91/42_works/minishell/srcs
	\$> cd srcs objs		ОК	UK	bash: cd: too many arguments \$>	1	/home/vietdu91/42_works/minishell
	\$> cd 'srcs'		ОК	ОК	\$>	0	/home/vietdu91/42_works/minishell/srcs
	\$> cd "srcs"		ОК	ОК	\$>	0	/home/vietdu91/42_works/minishell/srcs
	\$> cd '/etc'		ОК	ОК	\$>	0	Marche aussi avec les double quotes /
	\$> cd /e'tc'		ОК	ОК	\$>	Θ	/etc
	\$> cd /e"tc"		ОК	ОК	\$>	0	/etc
	\$> cd sr		ОК	ОК	bash: cd: sr: No such file or directory \$>	1	
	\$> cd Makefile		ОК		bash: cd: Makefile: Not a directory \$>	1	
	<pre>\$&gt; cd/minishell</pre>		ОК	ОК	\$>	0	/home/vietdu91/42_works/minishell
	\$> cd///		ОК	ОК	\$>	0	1
	\$> cd/.///bin/ls		ОК	ОК	<pre>bash: cd:/.///bin/ls: No such file or directory \$&gt;</pre>	1	/home/vietdu91/42_works/minishell
	\$> cd /		ОК	ОК	\$>	0	1
	\$> cd '/'		OK		\$>	Θ	Marche aussi avec les double quotes
	\$> cd //	NON GERE	OK	OK		0	<u>.</u> 1
	\$> pwd	NON GERE	ОК		\$>		
	\$> cd '//' \$> pwd	NON GERE	ОК	()K	// \$>	0	Marche aussi avec les double quotes /
	\$> cd /// \$> pwd		ОК	ОК	/ \$>	Θ	
	\$> cd /////// \$> pwd		ОК	ок	/ \$>	0	
	\$> cd '//////' \$> pwd		ОК	ОК	   \$>	0	Marche aussi avec les double quotes /
٥	\$> cd /minishell		ОК		<pre>bash: cd: /minishell: No such file or directory \$&gt;</pre>	1	
PWD	\$> cd / \$> cd		ОК	OK	\$>	0	1
© □	\$> cd _		ОК	UK	<pre>bash: cd: _: No such file or directory \$&gt;</pre>	1	
U	\$> cd -		ОК	ОК	/home/vietdu91/42_works/minishell \$>		Retourne à \$OLD_PWD \$OLD_PWD et \$PWD sont inversés /home/vietdu91/42_works/minishell
	\$> cd	NON GERE	ОК	ОК	\$>		Retourne à \$HOME \$OLD_PWD et \$PWD sont inversés /home/vietdu91

\$> cd		ОК	ок	bash: cd:: invalid option \$>	2	
\$> cd \$HOME		ОК	ОК	\$>	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>\$&gt; cd "\$PWD/srcs"</pre>		ОК	ОК	\$>	0	/home/vietdu91/42_works/minishell/srcs
\$> cd '\$PWD/srcs'		ОК	ок	<pre>bash: cd: \$PWD/srcs: No such file or directory \$&gt;</pre>	1	
<pre>\$&gt; unset HOME \$&gt; cd \$HOME</pre>		ОК	ОК	bash: cd: HOME not set \$>	1	
<pre>\$&gt; unset HOME \$&gt; export HOME= \$&gt; cd</pre>		ОК	ок	\$>	Θ	/home/vietdu91/42_works/minishell
<pre>\$&gt; unset HOME \$&gt; export HOME \$&gt; cd</pre>		ОК	ОК	bash: cd: HOME not set \$>	1	
<pre>\$&gt; cd minishell Docs crashtest.c</pre>		ОК	ок	bash: cd: too many arguments \$>	1	
<pre>\$&gt; cd /   echo \$? \$&gt; pwd</pre>		ОК	ОК	0 \$>	0 0	/home/vietdu91/42_works/minishell
\$> cd ~		ОК	OK	\$>	0	/home/vietdu91
\$> cd ~/	<b>NON GERE</b>	ОК	ОК	\$>	0	/home/vietdu91
\$> cd ~/   echo \$? \$> pwd	NON GERE	ОК	ОК	0 \$>	0	/home/vietdu91/42_works/minishell
\$> cd *	NON GERE	ОК	ОК	\$>	Θ	[BONUS] S'il n'y a qu'un seul <u>dossier</u> et rien d'autre
\$> cd *	NON GERE	ОК	ОК	<pre>bash: cd: minishell: No such file or directory \$&gt;</pre>	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>\$&gt; mkdir a \$&gt; mkdir a/b \$&gt; cd a/b \$&gt; rm -r//a \$&gt; cd</pre>		ок	ОК	<pre>chdir: error retrieving current directory: getcwd: cannot access parent directories: No such file or directory \$&gt;</pre>		
<pre>\$&gt; mkdir a \$&gt; mkdir a/b \$&gt; cd a/b \$&gt; rm -r//a \$&gt; pwd</pre>		ок	ОК	<pre>pwd: error retrieving current directory: getcwd: cannot access parent directories: No such file or directory \$&gt;</pre>	1	
<pre>\$&gt; mkdir a \$&gt; mkdir a/b \$&gt; cd a/b \$&gt; rm -r//a \$&gt; echo \$PWD \$&gt; echo \$OLDPWD</pre>		ок	ОК	/home/vietdu91/42_works/minishell/a/b \$> /home/vietdu91/42_works/minishell \$>	0 0	
<pre>\$&gt; mkdir a \$&gt; mkdir a/b \$&gt; cd a/b \$&gt; rm -r//a \$&gt; cd \$&gt; echo \$PWD \$&gt; echo \$OLDPWD</pre>		ОК	ОК	/home \$> /home/vietdu91/42_works/minishell/a/b \$>	0	
<pre>\$&gt; mkdir a \$&gt; cd a \$&gt; rm -r/a \$&gt; echo \$PWD \$&gt; echo \$OLDPWD</pre>		ОК	ОК	<pre>///home/vietdu91/42_works/minishell/a \$&gt;//home/vietdu91/42_works/minishell \$&gt;</pre>	0 0	

	<pre>\$&gt; export CDPATH=/ \$&gt; cd home/vietdu91</pre>	NON GERE	С	к ок	/home/vietdu91 \$>	0 0	A gerer ? /home/vietdu91
	<pre>\$&gt; export CDPATH=./ \$&gt; cd .</pre>	NON GERE	С	к ок	\$>	0	A gerer ? /home/vietdu91/42_works/minishell
	<pre>\$&gt; export CDPATH=./ \$&gt; cd</pre>	NON GERE	C	к ок	\$>	0 0	A gerer ? /home/vietdu91/42_works
	<pre>\$&gt; chmod 000 minishell \$&gt; ./minishell</pre>		C	ок	<pre>bash: ./minishell : Permission denied \$&gt;</pre>	126	
	\$> ls hola		С	ок ок	ls: cannot access 'hola': No such file or directory \$>	2	
	\$> ./Makefile		С	ок ок	<pre>bash: ./Makefile: Permission denied \$&gt;</pre>	126	
	\$> ./minishell		C	к ок	\$>	0	Relance le Minishell Il faut donc 2 fois exit pour exit
	<pre>\$&gt; env   grep SHLVL \$&gt; ./minishell \$&gt; env   grep SHLVL \$&gt; exit \$&gt; env   grep SHLVL</pre>	NON GERE	C	к ок	\$>	0 0 0 0	
	\$> touch hola \$> ./hola		C	КОК	bash: ./hola: Permission denied \$>	126	
<b>-</b> ✓	\$> env "wc" -l		C	КОК	66 \$>	0	
BÂTARDS	\$> env "wc "-l		C	к ок	bash: wc -l: command not found \$>	127	
BÂ	\$> expr 1 + 1		С	к ок	2 \$>	0	
	\$> expr \$? + \$?		C	к ок	0 \$>	0	Fait la somme de la valeur de \$?
	<pre>\$&gt; env -i ./minishell \$&gt; env</pre>		C	к ок	PWD=/mnt/nfs/homes/emtran/Desktop/42_minishell SHLVL=1 _=/usr/bin/env \$>	0	
	<pre>\$&gt; env -i ./minishell \$&gt; export</pre>		C	к ок	<pre>export OLDPWD export PWD="/mnt/nfs/homes/emtran/Desktop/42_minishell" export SHLVL="1" \$&gt;</pre>	0	
	<pre>\$&gt; env -i ./minishell \$&gt; cd</pre>		C	к ок	bash: cd: HOME not set \$>	1	
	<pre>\$&gt; env -i ./minishell \$&gt; cd ~</pre>	NON GERE	С	к ок	\$>	0	/home/vietdu91
	\$> exit		С	к ок	exit ~/minishell \$>	0	
	\$> exit exit		C	к ок	exit bash: exit: exit: numeric argument required ~/minishell \$>	2	
	\$> exit hola		C	ок	exit bash: exit: hola: numeric argument required ~/minishell \$>	2	
	\$> exit hola que tal		C	к ок	exit bash: exit: hola: numeric argument required ~/minishell \$>	2	
	\$> exit 42		С	ОК	exit ~/minishell \$>	42	
	\$> exit 000042		C	ОК	exit ~/minishell \$>	42	
	\$> exit 666		С	ок	exit ~/minishell \$>	154	
	\$> exit 666 666		C	к ок	exit bash: exit: too many arguments \$>	1	

\$> exit hola 666	ОК		hach: ovit: too many acquiments		
\$> exit hola 666			<pre>bash: exit: too many arguments \$&gt;</pre>		
	ОК	ок	<pre>exit bash: exit: hola: numeric argument required ~/minishell \$&gt;</pre>	2	
\$> exit 666 666 666	ок	ОК	exit bash: exit: too many arguments \$>	1	
\$> exit 666 hola 666	ок	OK		1	
\$> exit hola 666 666	ок	ОК		2	
\$> exit 259	ок	OK		3	
\$> exit -4	ок	OK	exit ~/minishell \$>	252	
\$> exit -42	ок		exit ~/minishell \$>	214	
\$> exit -0000042	ок		exit ~/minishell \$>	214	
\$> exit -259	ок	ОК	exit ~/minishell \$>	253	
\$> exit -666	ок		exit ~/minishell \$>	102	
\$> exit +666	ок		exit ~/minishell \$>	154	
\$> exit 0	ок		exit ~/minishell \$>	0	
\$> exit +0	ок		exit ~/minishell \$>	0	
\$> exit -0	ок	ок	exit ~/minishell \$>	0	
\$> exit +42	ок	ОК	exit ~/minishell \$>	42	
\$> 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		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	ОК	exit ~/minishell \$>	154	
\$> exit '666'	ок	ОК	exit ~/minishell \$>	154	
\$> exit '-666'	ок	ОК	exit ~/minishell \$>	102	Marche aussi pour les double quotes
\$> exit '+666'	ок	ОК	exit ~/minishell \$>	154	Marche aussi pour les double quotes
\$> exit '666'	ОК	ОК	exit bash: exit:666: numeric argument required ~/minishell \$>	2	Marche aussi pour les double quotes

\$> exit '++++666'	ОК	ОК	exit bash: exit: ++++666: numeric argument required ~/minishell \$>	2	Marche aussi pour les double quotes
\$> exit '6'66	ок		exit ~/minishell \$>	154	Marche aussi pour les double quotes
\$> exit '2'66'32'	ОК	ок	exit ~/minishell \$>	8	Marche aussi pour les double quotes
\$> exit "'666'"	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 '666'"666"666	ОК		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	OK	exit bash: exit: 9223372036854775808: numeric argument required ~/minishell \$>	2	
\$> exit -9223372036854775808	ОК		exit ~/minishell \$>	Θ	
\$> exit -9223372036854775809	ОК	OK	exit bash: exit: -9223372036854775809: numeric argument required ~/minishell \$>	2	
\$> cat   cat   ls	ок	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$&gt;</pre>	0	Apparaît la ligne ls Et 3 retours à la ligne avec la touche Entrée
\$> ls   exit	ок	ОК	\$>	0	Ne fait rien  Chaque commande d'un pipeline est exécutée dans son propre sous-shell, qui est un processus distinct
\$> ls   exit 42	ОК	OK	\$>	42	Ne fait rien
\$> exit   ls	ОК	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$&gt;</pre>	0	
<pre>\$&gt; echo hola &gt; bonjour \$&gt; exit   cat -e bonjour</pre>	ОК	UK	hola\$ \$>	0	
<pre>\$&gt; echo hola &gt; bonjour \$&gt; cat -e bonjour   exit</pre>	ОК	ОК	\$>	0	
\$> echo   echo	ОК	ОК	\$>	0	
\$> echo hola   echo que tal	ОК	UK	que tal \$>	0	
\$> pwd   echo hola	ОК	OK	hola \$>	0	
\$> env   echo hola	ОК	OK	hola \$>	0	
\$> echo oui   cat -e	ОК	OK	oui\$ \$>	0	
\$> echo oui   echo non   echo hola   grep oui	ОК		\$>	0	
\$> echo oui   echo non   echo hola   grep non	ОК		\$>	0	

	\$> echo oui   echo non   echo hola   grep hola	ОК		hola \$>	0	
	\$> echo hola   cat -e   cat -e   cat -e	ОК	ок	hola\$\$\$ \$>	0	
	\$> cd   echo "hola"	ОК	UK	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
	\$> ifconfig   grep ":"	ОК	ок	<pre>docker0: flags=4099<up,broadcast,multicast> mtu 1500   ether 02:42:27:cf:95:49 txqueuelen 0 (Ethernet) [] \$&gt;</up,broadcast,multicast></pre>	Θ	
	\$> ifconfig   grep hola	ОК		\$>	1	
	\$> whoami   grep \$USER	ОК	ок	vietdu91 \$>	Θ	
	<pre>\$&gt; whoami   grep \$USER &gt; /tmp/bonjour \$&gt; cat /tmp/bonjour</pre>	ОК	ок	vietdu91 \$>	0	
	<pre>\$&gt; whoami   cat -e   cat -e &gt; /tmp/bonjour \$&gt; cat /tmp/bonjour</pre>	ОК	ок	vietdu91\$\$ \$>	Θ	
	<pre>\$&gt; whereis ls   cat -e   cat -e &gt; /tmp/bonjour \$&gt; 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	
<u>_</u>	\$> 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	<pre>bash: hola: command not found crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$&gt;</pre>	Θ	
	\$> ls   ls   hola   rev	ОК	ок	<pre>bash: hola: command not found \$&gt;</pre>	Θ	
	\$> ls   ls   echo hola   rev	ОК	ок	aloh \$>	0	
	\$> ls -la   grep "."	ОК	ОК	total 16 drwx 2 vietdu91 [] . drwxrwxewt 51 root []rwxr-xr-x 8 vietdu91 [] crashtest.c [] \$>	Θ	Sort la commande entiere de ls -la
	\$> ls -la   grep "'.'"	OK	ОК	\$>	Θ	
	<pre>\$&gt; 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		test.c\$	Θ	
	<pre>\$&gt; ls   s ls  ls</pre>	ОК	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$&gt;</pre>	Θ	
	\$> echo hola   cat   cat   cat   cat   grep hola	ОК		hola \$>	0	
	\$> echo hola   cat	ОК		hola \$>	0	
	\$> echo hola  cat	ОК		hola \$>	0	
	\$> echo hola  cat	ОК	OK	hola \$>	0	
	\$> echo hola cat	ок	ОК	hola \$>	0	

<pre>\$&gt; echo hola    cat</pre>	NON GERE	ОК	ОК	hola \$>	0	[BONUS]
\$> echo hola     cat		ок		bash: syntax error near unexpected token ` ' \$>	2	
<pre>\$&gt; ech o hola   cat</pre>		ок		bash: ech: command not found bash: o: command not found \$>	127	
<pre>\$&gt; cat Makefile   cat -e   cat -e</pre>		ок	ОК	[] \$\$ .PHONY: all clean fclean re\$\$ \$>	0	
<pre>\$&gt; cat Makefile   grep srcs   cat -e</pre>		ОК	ОК	objs/%.o: srcs/%.c\$ @norminette srcs/*\$ \$>	0	
<pre>\$&gt; cat Makefile   grep srcs   grep srcs   cat -e</pre>		ОК	ОК	objs/%.o: srcs/%.c\$ @norminette srcs/*\$ \$>	0	
<pre>\$&gt; 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 \$&gt;</pre>	1	
\$> cat Makefile   grep pr   head -n 5   hello		ОК	ОК	bash: hello: command not found \$>	127	
<pre>\$&gt; export HOLA=bonjour   cat -e   cat -e</pre>		ОК	ОК	\$>	0	
\$> unset HOLA   cat -e		ОК	ОК	\$>	Θ	
<pre>\$&gt; export HOLA   echo hola \$&gt; env   grep PROUT</pre>		ОК	ОК	hola \$> \$>	0	
<pre>\$&gt; export   echo hola</pre>		ОК	ОК	hola \$>	0	
\$> sleep 3   sleep 3		ОК	ОК	\$>	0	Le prompt apparait au bout de 3 secondes
<pre>\$&gt; time sleep 3   sleep 3</pre>		ОК	OK	real 0m3.002s user 0m0.003s sys 0m0.001s \$>	0	Le prompt apparait au bout de 3 secondes
<pre>\$&gt; sleep 3   exit</pre>		ОК	ОК	\$>	0	Le prompt apparait au bout de 3 secondes
\$> exit   sleep 3		ОК	ОК	\$>	0	Le prompt apparait au bout de 3 secondes
<pre>\$&gt; echo hola &gt; a \$&gt; &gt;&gt;b echo que tal \$&gt; cat a   <b cat=""  =""> c   cat</b></pre>		ОК	ОК	\$>	0	
\$> pwd && ls	NON GERE	ок	ОК	<pre>/home/vietdu91/42_works/minishell crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$&gt;</pre>	0	[BONUS]
\$> pwd    ls	NON GERE	ок	ОК	<pre>/home/vietdu91/42_works/minishell \$&gt;</pre>	0	[BONUS]
<pre>\$&gt; echo hola    echo bonjour</pre>	NON GERE	ОК	ОК	hola \$>	0	[BONUS]
<pre>\$&gt; echo hola &amp;&amp; echo bonjour</pre>	NON GERE	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	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 0 hola \$>	[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 \$>	[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 0 philosophers srcs test.c \$&gt;</pre>	[BONUS]
=	\$> export ""    ls	NON GERE	ОК	ОК	<pre>bash: export: `': not a valid identifier crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$&gt;</pre>	[BONUS]
88	\$> ls && export ""	NON GERE	ОК	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell 1 philosophers srcs test.c bash: export: `': not a valid identifier \$&gt;</pre>	[BONUS]
	\$> export "" && ls	NON GERE	ОК	ОК	bash: export: `': not a valid identifier 1 \$>	[BONUS]
	\$> ls    ;	NON GERE	ОК		bash: syntax error near unexpected token `;' \$>	[BONUS] Marche pour les metacharacters  &;()
	\$> ;    ls	NON GERE	ОК		bash: syntax error near unexpected token `;' \$>	[BONUS] Marche pour les metacharacters  &;()
	\$> ls && ;	NON GERE	ОК		<pre>bash: syntax error near unexpected token `;' \$&gt;</pre>	[BONUS] Marche pour les metacharacters  &;()
	\$> ; && ls	NON GERE	ОК		bash: syntax error near unexpected token `;' \$>	[BONUS] Marche pour les metacharacters  &;()
	\$> ls    <	NON GERE	ОК	ОК	bash: syntax error near unexpected token `newline' 2 \$>	[BONUS] Marche pour les metacharacters <>
	\$> ls && <	NON GERE	ОК	ОК	<pre>bash: syntax error near unexpected token `newline' \$&gt;</pre>	[BONUS] Marche pour les metacharacters <>
	\$> cat   echo    ls	NON GERE	ОК	ОК	\$>	[BONUS]
	\$> cat   echo && ls	NON GERE	ок	ОК	crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$>	[BONUS]
	\$> ls    cat   echo	NON GERE	ОК	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell 0 philosophers srcs test.c \$&gt;</pre>	[BONUS]
	\$> ls && cat   echo	NON GERE	ОК	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell 0 philosophers srcs test.c \$&gt;</pre>	[BONUS]
	\$> export "" && unset ""	NON GERE	ОК	ОК	bash: export: `': not a valid identifier 1	[BONUS]
	\$> (ls)	NON GERE	ОК	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell 0 philosophers srcs test.c \$&gt;</pre>	[BONUS]
	\$> ( ( ls ) )	NON GERE	ОК	ОК	<pre>crashtest.c Docs Dockers Drawings Makefile minishell 0 philosophers srcs test.c \$&gt;</pre>	[BONUS]
	\$> ( ( ) ls )	NON GERE	ОК		bash: syntax error near unexpected token `)' \$>	[BONUS]

\$> ls && (ls)				crashtest.c Docs Dockers Drawings Makefile minishell	0	[BONUS]
	NON GERE	ок	ОК	philosophers srcs test.c crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c \$>		
\$> (ls && pwd)	NON GERE	ок	UK	<pre>crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c /home/vietdu91/42_works/minishell \$&gt;</pre>	Θ	[BONUS]
\$> ( ( ls&&pwd ) )	NON GERE	ОК		crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c /home/vietdu91/42_works/minishell	0	[BONUS]
\$> ( ( ls ) &&pwd )	NON GERE	ОК	ОК	crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c /home/vietdu91/42_works/minishell	0	[BONUS]
\$> (ls && ( ( pwd ) ) )	NON GERE	ОК	ОК	crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c /home/vietdu91/42_works/minishell	0	[BONUS]
<pre>\$&gt; (ls &amp;&amp; pwd) &gt; hola \$&gt; cat hola</pre>	NON GERE	ОК	ОК	crashtest.c Docs Dockers Drawings Makefile minishell philosophers srcs test.c /home/vietdu91/42_works/minishell \$>	Θ	[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]
<pre>\$&gt; (&gt; pwd) \$&gt; ls</pre>	NON GERE	ОК	OK	crashtest.c Docs Dockers Drawings Makefile minishell philosophers pwd srcs test.c \$>	0	[BONUS]
\$> (< pwd) \$> ls	NON GERE	ОК	ОК	\$>	0	[BONUS] Si pwd existe
\$> (< pwd)	NON GERE	ок	ОК	bash: pwd: No such file or directory \$>	1	[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 `)' \$>	2	[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 philosophers pwd srcs test.c crashtest.c Docs Dockers Drawings Makefile minishell philosophers pwd srcs test.c \$>	Θ	[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 \$>	Θ	[BONUS]
\$> (lss&&pwd)&&(lss&&pwd)	NON GERE	ок	()K	bash: lss: command not found \$>	127	[BONUS]

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

<pre>\$&gt; (ls &gt; Docs/hey &amp;&amp; pwd) &gt; hola \$&gt; cat hola \$&gt; cat Docs/hey</pre>	NON GERE	ок		<pre>/home/vietdu91/42_works/minishell \$&gt; crashtest.c Docs [] srcs test.c \$&gt;</pre>	0 0 0	[BONUS]
<pre>\$&gt; ls &gt; Docs/hey &amp;&amp; pwd &gt; hola \$&gt; cat hola \$&gt; 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	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 \$> pwd	NON GERE	ОК	ок	ls: invalid option 'z' /home/vietdu91 \$> /home/vietdu91 \$>	0 0	[BONUS]
\$> ls -z    (cd//&&pwd) \$> pwd	NON GERE	ОК	ОК	<pre>ls: invalid option 'z' /home/vietdu91 \$&gt; /home/vietdu91/42_works/minishell \$&gt;</pre>	0 0	[BONUS]
<pre>\$&gt; echo hola &gt; bonjour \$&gt; cat bonjour</pre>		ОК	ОК	hola \$>	0	
<pre>\$&gt; echo que tal &gt;&gt; bonjour \$&gt; cat bonjour</pre>		ОК		hola que tal \$>	0 0	
<pre>\$&gt; echo hola &gt; bonjour \$&gt; echo que tal &gt;&gt; bonjour \$&gt; cat &lt; bonjour</pre>		ок	OK	\$> \$> hola que tal \$>	0 0 0	
<pre>\$&gt; echo hola &gt; bonjour \$&gt; rm bonjour \$&gt; echo que tal &gt;&gt; bonjour \$&gt; cat &lt; bonjour</pre>		ОК	ок	que tal \$>	0 0 0 0	
<pre>\$&gt; echo hola que tal &gt; bonjour \$&gt; cat bonjour</pre>		ОК	ок	hola que tal \$>	0	
<pre>\$&gt; echo hola que tal &gt; /tmp/bonjour \$&gt; cat -e /tmp/bonjour</pre>		ОК	ок	hola que tal\$ \$>	0	
<pre>\$&gt; export HOLA=hey \$&gt; echo bonjour &gt; \$HOLA \$&gt; echo \$HOLA</pre>		ОК	ок	\$> hola \$>	0 0 0	Ne fait rien
<pre>\$&gt; whereis grep &gt; Docs/bonjour \$&gt; cat Docs/bonjour</pre>		ОК	OK	grep: /usr/bin/grep /usr/share/man/man1/grep.1.gz /usr/share/man/man1/grep.1posix.gz /usr/share/info/grep.info.gz \$>	0 0	
<pre>\$&gt; ls -la &gt; Docs/bonjour \$&gt; cat Docs/bonjour</pre>		ок	ок	total 16 drwx 2 vietdu91 [] . drwxrwxewt 51 root []rwxr-xr-x 8 vietdu91 [] crashtest.c [] \$>	0 0	
<pre>\$&gt; pwd&gt;bonjour \$&gt; cat bonjour</pre>		ОК	ОК	<pre>/home/vietdu91/42_works/minishell \$&gt;</pre>	0	

<pre>\$&gt; pwd &gt; bonjour \$&gt; cat bonjour</pre>	ОК	ок	<pre>/home/vietdu91/42_works/minishell \$&gt;</pre>	
<pre>\$&gt; echo hola &gt; &gt; bonjour</pre>	ок	ок	bash: syntax error near unexpected token `>' 2 \$>	
\$> echo hola < < bonjour	ок	ок	bash: syntax error near unexpected token `<' 2 \$>	
\$> echo hola >>> bonjour	ОК	ОК	bash: syntax error near unexpected token `>' 2 \$>	
<pre>\$&gt; &gt; bonjour echo hola \$&gt; cat bonjour</pre>	ок	ОК	hola 0 \$> 0	
\$> > bonjour   echo hola \$> cat bonjour	ОК	ок	hola 0 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	
<pre>\$&gt; prout hola &gt; bonjour \$&gt; ls</pre>	ок	ок	bash: prout: command not found 127 \$> 0 bonjour []	Erreur mais file "bonjour" cree tout de meme
<pre>\$&gt; echo hola &gt; hello &gt;&gt; hello \$&gt; ls \$&gt; cat hello</pre>	ОК	ОК	\$> 0 hello 0 \$> 0 hola \$>	
<pre>\$&gt; echo hola &gt; hello &gt;&gt; hello \$&gt; echo hola &gt;&gt; hello \$&gt; cat &lt; hello</pre>	ок	ок	hola 0 0 5> 0	
<pre>\$&gt; echo hola &gt; hello &gt;&gt; hello \$&gt; echo hola &gt;&gt; hello \$&gt; echo hola &gt; hello &gt;&gt; hello &gt;&gt; hello \$&gt; cat &lt; hello</pre>	ОК	ОК	hola	
<pre>\$&gt; echo hola &gt;&gt; hello &gt;&gt; hello &gt; hello \$&gt; echo hola &gt;&gt; hello \$&gt; cat &lt; hello</pre>	ОК	ок	hola 0 0 0 5> 0	
<pre>\$&gt; echo hola &gt;&gt; hello &gt;&gt; hello &gt; hello \$&gt; echo hola &gt;&gt; hello \$&gt; echo hola &gt;&gt; hello &gt;&gt; hello &gt;&gt; hello \$&gt; cat &lt; hello</pre>	ок	ОК	hola	
<pre>\$&gt; echo hola &gt; hello \$&gt; echo hola &gt;&gt; hello &gt;&gt; hello \$&gt; echo hola &gt;&gt; hello \$&gt; cat &lt; hello</pre>	ок	ОК	hola 0 hola 0 hola 0 \$>> 0	
<pre>\$&gt; echo hola &gt; hello \$&gt; echo hey &gt; bonjour \$&gt; echo <bonjour <hello<="" pre=""></bonjour></pre>	ок	ок	\$> 0 0 0 0	
<pre>\$&gt; echo hola &gt; hello \$&gt; echo hey &gt; bonjour \$&gt; echo <hello <bonjour<="" pre=""></hello></pre>	ок	ок	\$> 0 0 0 0	
<pre>\$&gt; echo hola &gt; bonjour \$&gt; echo hey &gt; hello \$&gt; rm bonjour hello \$&gt; echo hola &gt; bonjour &gt; hello &gt; bonjour \$&gt; cat bonjour \$&gt; cat hello</pre>	ОК	ОК	hola	
<pre>\$&gt; echo hola &gt; bonjour \$&gt; echo hey &gt; hello \$&gt; echo hola &gt; bonjour &gt; hello &gt; bonjour \$&gt; cat bonjour \$&gt; cat hello</pre>	ОК	ок	hola	
<pre>\$&gt; echo hola &gt; bonjour \$&gt; echo hey &gt; hello \$&gt; rm bonjour hello \$&gt; echo hola &gt; bonjour &gt;&gt; hello &gt; bonjour \$&gt; cat bonjour \$&gt; cat hello</pre>	ОК	ОК	hola	

Section   Sect						
Company   Tenting   Tent	<pre>\$&gt; echo hey &gt; hello \$&gt; echo hola &gt; bonjour &gt; hello &gt; bonjour \$&gt; cat bonjour</pre>	ОК		\$>	0 0 0	
See the first proper in bettle or besteror control of the first part of the first pa	<pre>\$&gt; echo hey &gt; hello \$&gt; rm bonjour hello \$&gt; echo hola &gt; bonjour &gt; hello &gt;&gt; bonjour \$&gt; cat bonjour</pre>	ОК		\$>	0 0 0 0 0	
Section system   Section	<pre>\$&gt; echo hey &gt; hello \$&gt; echo hola &gt; bonjour &gt; hello &gt;&gt; bonjour \$&gt; cat bonjour</pre>	ОК		\$>		
Comparison   Com	<pre>\$&gt; echo hey &gt; hello \$&gt; rm bonjour hello \$&gt; echo hola &gt;&gt; bonjour &gt; hello &gt; bonjour \$&gt; cat bonjour</pre>	ОК		\$>		
Comparison   Com	<pre>\$&gt; echo hey &gt; hello \$&gt; echo hola &gt;&gt; bonjour &gt; hello &gt; bonjour \$&gt; cat bonjour</pre>	ОК		\$>	0 0 0 0	
OK OK OK   No.	<pre>\$&gt; echo hey &gt; hello \$&gt; rm bonjour hello \$&gt; echo hola &gt;&gt; bonjour &gt;&gt; hello &gt;&gt; bonjour \$&gt; cat bonjour</pre>	ОК		\$>	0 0 0 0 0	
S	<pre>\$&gt; echo hey &gt; hello \$&gt; echo hola &gt;&gt; bonjour &gt;&gt; hello &gt;&gt; bonjour \$&gt; cat bonjour</pre>	ОК		hola \$> hey	0 0 0 0	
Note		ОК	ок		0 0	
Secho Pello > hola2	hola>bonjour>hola>>bonjour>hola hey >bonjour hola >hola \$> cat bonjour	ОК		\$> hey hola		
\$> echo hello > hola2 \$> echo 2 >>hola1 > hola2 \$> locat hola1 \$> cat hola2 \$> pwd \$> pwd \$> locat hola2 \$> locat hola2 \$> locat hola2 \$> pwd \$> locat hola2 \$> locat	<pre>\$&gt; echo hello &gt; hola2 \$&gt; echo 2 &gt;hola1 &gt;&gt; hola2 \$&gt; ls \$&gt; cat hola1</pre>	ОК		<pre>\$&gt; \$&gt; hello 2</pre>	0 0 0 0	
\$> \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	<pre>\$&gt; echo hello &gt; hola2 \$&gt; echo 2 &gt;&gt;hola1 &gt; hola2 \$&gt; ls \$&gt; cat hola1</pre>	ОК		\$> bonjour \$> 2	0 0 0 0	
\$> < Makefile .  NON GERE  OK  OK  OK  OK  S>  Inot enough arguments S>  Cat < pwd  OK  OK  Dash: pwd: No such file or directory  1		ОК	ОК		· ·	
\$> cat <pwd< td=""><td>\$&gt; &lt; pwd</td><td>ОК</td><td>ОК</td><td></td><td>1</td><td></td></pwd<>	\$> < pwd	ОК	ОК		1	
	\$> < Makefile . NON GERE	ОК	ОК		1	Voir pourquoi
	\$> cat <pwd< td=""><td>ОК</td><td>ОК</td><td></td><td>1</td><td></td></pwd<>	ОК	ОК		1	

\$> cat <srcs pwd<="" th=""><th>ОК</th><th>ок</th><th>bash: srcs/pwd: No such file or directory \$&gt;</th><th>1</th><th></th></srcs>	ОК	ок	bash: srcs/pwd: No such file or directory \$>	1	
\$> cat <td>ок</td> <td>ок</td> <td><pre>bash:/pwd: No such file or directory \$&gt;</pre></td> <td>1</td> <td></td>	ок	ок	<pre>bash:/pwd: No such file or directory \$&gt;</pre>	1	
\$> cat >>	ОК	ок	bash: syntax error near unexpected token `newline' \$>	2	Marche aussi pour les autres operateurs de redirection
\$> cat >>>	ок	ок	<pre>bash: syntax error near unexpected token `&gt;' \$&gt;</pre>	2	
\$> cat >> <<	ок	ок	bash: syntax error near unexpected token `<<' \$>	2	
\$> 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 \$> ls	ОК	ок	bash: ls2: No such file or directory \$> ls1 [] \$>	1 0	
\$> >>hola \$> cat hola	ок	ОК	\$>	0 0	
\$> echo hola > bonjour \$> cat < bonjour	ок	UK	hola \$>	0 0	
\$> echo hola >bonjour \$> cat <bonjour< td=""><td>ок</td><td>UK</td><td>hola \$&gt;</td><td>0 0</td><td></td></bonjour<>	ок	UK	hola \$>	0 0	
\$> echo hola>bonjour \$> cat <bonjour< td=""><td>ок</td><td>UK</td><td>hola \$&gt;</td><td>0 0</td><td></td></bonjour<>	ок	UK	hola \$>	0 0	
\$> echo hola> bonjour \$> cat< bonjour	ОК	UK	hola \$>	0 0	
\$> echo hola	ок	UK	hola \$>	0 0	
\$> echo hola > bonjour \$> cat < bonjour	ок	UK	hola \$>	0 0	
\$> echo hola > srcs/bonjour \$> cat < srcs/bonjour	ок	ОК	hola \$>	0 0	
<pre>\$&gt; echo hola &gt;srcs/bonjour \$&gt; cat <srcs bonjour<="" pre=""></srcs></pre>	ок	UK	hola \$>	0 0	
\$> echo hola > bonjour \$> echo que tal >> bonjour \$> cat < bonjour	ОК	ОК	hola que tal \$>	0 0 0	
\$> cat < bonjour  \$> echo hola > bonjour  \$> rm bonjour  \$> echo que tal >> bonjour  \$> cat < bonjour	ОК		que tal \$>	0 0 0 0	
\$> cat < bonjour  \$> e'c'"h"o hola > bonjour  \$> cat 'bo'"n"jour	ок	ок	hola \$>	0 0	
<pre>\$&gt; echo hola &gt; bonjour\ 1  \$&gt; ls  \$&gt; cat bonjour\ 1</pre> NON GERE	ОК	ок	bonjour 1' [] \$> hola \$>	0 0 0	
<pre>\$&gt; echo hola &gt; bonjour hey \$&gt; ls \$&gt; cat bonjour \$&gt; cat hey</pre>	ОК	ОК	bonjour [] \$> hola hey \$> cat: hey: No such file or directory \$>	0 0 0 1	
<pre>\$&gt; echo hola &gt; srcs/bonjour \$&gt; echo hey &gt; srcs/hello \$&gt; &gt;srcs/bonjour &gt;srcs/hello <prout \$=""> cat srcs/bonjour srcs/hello</prout></pre>	ОК	ОК	<pre>bash: prout: No such file or directory \$&gt; \$&gt;</pre>	0 0 1 0	

\$> echo hola > srcs/bonjour			bash: prout: No such file or directory 0
<pre>\$&gt; echo hey &gt; srcs/hello  \$&gt; rm srcs/bonjour srcs/hello</pre>			\$> 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
\$> >srcs/bonjour >srcs/hello <prout< td=""><td>ОК</td><td>ОК</td><td>\$&gt; 1</td></prout<>	ОК	ОК	\$> 1
V \$> ls srcs			\$>
\$> cat srcs/bonjour srcs/hello			
<pre>\$&gt; echo hola &gt; srcs/bonjour \$&gt; echo hey &gt; srcs/hello</pre>			bash: prout: No such file or directory 0
<pre>\$&gt; &gt;srcs/bonjour <prout>srcs/hello</prout></pre>	ОК	ок	\$>
\$> cat srcs/bonjour			hey 0
\$> cat srcs/hello			
<pre>\$&gt; echo hola &gt; srcs/bonjour \$&gt; echo hey &gt; srcs/hello</pre>			bash: prout: No such file or directory 0 \$> 0
<pre>\$&gt; rm srcs/bonjour srcs/hello</pre>	ОК	ОК	bonjour 0
<pre>\$&gt; &gt;srcs/bonjour <prout>srcs/hello \$&gt; ls srcs</prout></pre>	O IX	O K	\$> 5
\$> ts sics \$> cat srcs/bonjour			\$> 0
\$> echo hola >/bonjour			bash: prout: No such file or directory 0
\$> echo hey >/hello	ОК	ОК	\$>
<pre>\$&gt; &gt;/bonjour &gt;/hello <pre>cat/bonjour/hello</pre></pre>			\$> 0
\$> echo hola >/bonjour			bash: prout: No such file or directory 0
\$> echo hey >/hello			\$>
<pre>\$&gt; rm/bonjour/hello \$&gt; &gt;/bonjour &gt;/hello <prout< pre=""></prout<></pre>	ОК	ок	bonjour hello  o
\$> ls			\$> \$>
<pre>\$&gt; cat/bonjour/hello</pre>			Θ
\$> echo hola >/bonjour			bash: prout: No such file or directory
<pre>\$&gt; echo hey &gt;/hello \$&gt; &gt;/bonjour <prout>/hello</prout></pre>	ОК	ОК	\$> \$>
\$> cat/bonjour			hey 0
\$> cat/hello			\$>
<pre>\$&gt; echo hola &gt;/bonjour \$&gt; echo hey &gt;/hello</pre>			bash: prout: No such file or directory 0 \$> 0
<pre>\$&gt; rm/bonjour/hello</pre>	ОК	ок	bonjour 0
<pre>\$&gt; &gt;/bonjour <prout>/hello</prout></pre>	UK	OK .	\$>
<pre>\$&gt; ls \$&gt; cat/bonjour</pre>			\$> 0
\$> echo hola > srcs/bonjour			bash: prout: No such file or directory 0
<pre>\$&gt; echo hey &gt; srcs/hello</pre>			\$>
<pre>\$&gt; &gt;srcs/bonjour &gt;&gt;srcs/hello <pre>prout \$&gt; cat srcs/bonjour</pre></pre>	ОК	ОК	\$> 1 0
\$> cat srcs/hello			hey
\$> echo hola > srcs/bonjour			bash: prout: No such file or directory 0
<pre>\$&gt; echo hey &gt; srcs/hello</pre>	OK	016	\$>
<pre>\$&gt; &gt;&gt;srcs/bonjour &gt;srcs/hello <prout \$=""> cat srcs/bonjour</prout></pre>	ОК	ОК	bonjour 1 0
\$> cat srcs/hello			\$>
\$> echo hola > srcs/bonjour			bash: prout: No such file or directory 0
<pre>\$&gt; echo hey &gt; srcs/hello \$&gt; &gt;&gt;srcs/bonjour &gt;&gt;srcs/hello <prout< pre=""></prout<></pre>			\$> bonjour 1
<pre>\$&gt; cat srcs/bonjour</pre>	ОК	ОК	\$>
<pre>\$&gt; cat srcs/hello</pre>			hey 0
fire asks help a coss/bosines			
<pre>\$&gt; echo hola &gt; srcs/bonjour \$&gt; echo hey &gt; srcs/hello</pre>			bash: prout: No such file or directory 0 \$> 0
<pre>\$&gt; &gt;srcs/bonjour <prout>&gt;srcs/hello</prout></pre>	ОК	ок	\$> 1
<pre>\$&gt; cat srcs/bonjour \$&gt; cat srcs/hello</pre>			hey c
\$> cat srcs/netto  \$> echo hola > srcs/bonjour			bash: prout: No such file or directory 0
<pre>\$&gt; echo hey &gt; srcs/hello</pre>			\$> 0
<pre>\$&gt; &gt;&gt;srcs/bonjour <prout>srcs/hello</prout></pre>	ОК	ОК	bonjour 1
<pre>\$&gt; cat srcs/bonjour \$&gt; cat srcs/hello</pre>			\$>
and the street of the street o			\$>

<pre>\$&gt; echo hola &gt; srcs/bonjour \$&gt; echo hey &gt; srcs/hello \$&gt; &gt;&gt;srcs/bonjour <prout>&gt;srcs/hello \$&gt; cat srcs/bonjour \$&gt; cat srcs/hello</prout></pre>	ОК	ОК	bash: prout: No such file or directory \$> 0 bonjour 1 \$> 0 hey \$>
<pre>\$&gt; echo hola &gt; srcs/bonjour \$&gt; echo hey &gt; srcs/hello \$&gt; <prout>&gt;srcs/bonjour &gt;&gt;srcs/hello \$&gt; cat srcs/bonjour \$&gt; cat srcs/hello</prout></pre>	OK	ОК	<pre>bash: prout: No such file or directory \$&gt;</pre>
<pre>\$&gt; echo hola &gt; bonjour \$&gt; echo hey &gt; hello \$&gt; <bonjour>hello \$&gt; cat bonjour \$&gt; cat hello</bonjour></pre>	ОК	ОК	hola \$> 0 0 0 0 0 0 0 0 0 0
<pre>\$&gt; echo hola &gt; bonjour \$&gt; echo hey &gt; hello \$&gt; &gt;bonjour &gt;hello &lt; prout \$&gt; cat bonjour \$&gt; cat hello</pre>	ОК	ОК	bash: prout: No such file or directory  \$> 5> 1 \$> 0 0 0 0
<pre>\$&gt; echo hola &gt; bonjour \$&gt; echo hey &gt; hello \$&gt; rm bonjour hello \$&gt; &gt;bonjour &gt;hello &lt; prout \$&gt; cat bonjour \$&gt; cat hello</pre>	ок	ок	bash: prout: No such file or directory  \$> 0 \$> 0 \$> 1 0 0 0
<pre>\$&gt; echo hola &gt; bonjour \$&gt; echo hey &gt; hello \$&gt; &gt;bonjour <pre>\$&gt; cat bonjour \$&gt; cat hello</pre></pre>	ОК	ОК	bash: prout: No such file or directory  \$>
<pre>\$&gt; echo hola &gt; bonjour \$&gt; echo hey &gt; hello \$&gt; rm bonjour hello \$&gt; &gt;bonjour <pre>prout hello \$&gt; cat bonjour</pre></pre>	ОК	ОК	bash: prout: No such file or directory  \$> 0 5> 0 1 0
<pre>\$&gt; echo hola &gt; bonjour \$&gt; <bonjour cat="" wc=""  =""> bonjour \$&gt; cat bonjour</bonjour></pre>	ок	ок	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
\$> rm -f bonjour \$> rm bonjour > bonjour \$> ls -l bonjour	ОК	ок	ls: cannot access 'bonjour': No such file or directory 0 0 2
<pre>\$&gt; export HOLA="bonjour hello" \$&gt; &gt;\$HOLA \$&gt; ls</pre>	ОК	ОК	'bonjour hello' [] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
<pre>\$&gt; export HOLA="bonjour hello" \$&gt; &gt; "\$HOLA" \$&gt; ls</pre>	ОК	ок	'bonjour hello' [] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
<pre>\$&gt; export HOLA="bonjour hello" \$&gt; &gt;\$"HOLA" \$&gt; ls</pre>	ОК	ок	HOLA [] \$> 0 0 0
<pre>\$&gt; export HOLA="bonjour hello" \$&gt; &gt;\$HOLA&gt;hey \$&gt; ls</pre>	ОК	ок	'bonjour hello' hey [] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
\$> export HOLA="bonjour hello" \$> >hey>\$HOLA \$> ls	ОК	ОК	'bonjour hello' hey [] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
<pre>\$&gt; export HOLA="bonjour hello" \$&gt; &gt;hey&gt;\$HOLA&gt;hey&gt;hey \$&gt; ls</pre>	ОК	ок	'bonjour hello' hey [] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

	_					_	
<pre>\$&gt; export A=hey \$&gt; export A B=Hola D E C="Que Tal"</pre>					\$> \$>	0 0	
<pre>\$&gt; echo \$PROUT\$B\$C &gt; /tmp/a &gt; /tmp/b &gt; /tmp/c \$&gt; cat /tmp/a</pre>			ОК	ОК	\$> HolaQue Tal	0 0	
<pre>\$&gt; cat /tmp/b \$&gt; cat /tmp/c</pre>					\$>	0 0	
(I)	LI Contenu du f						
111	!! Contenu du f !! Contenu du f						
\$> <a <b="" <c<="" cat="" td=""><td></td><td></td><td>ОК</td><td>ок</td><td>Prout \$&gt;</td><td>0</td><td></td></a>			ОК	ок	Prout \$>	0	
\$> <a <b="" <c<="" cat="" td=""><td></td><td></td><td></td><td></td><td>\$&gt;</td><td>0</td><td></td></a>					\$>	0	
<pre>\$&gt; cat a \$&gt; cat b</pre>			Olí	0.14	\$> \$>	0	
\$> cat c			OK		Amour Tu es Horrible 0123456789	0	
					Prout \$>		
<pre>\$&gt; &gt;a ls &gt;b &gt;&gt;c &gt;d \$&gt; cat a</pre>					\$> \$>	0	
\$> cat b \$> cat c					\$> Prout	0 0	
\$> cat d			ОК	OK	\$> crashtest.c	0	
					Docs []		
					test.c		
\$> >a ls >b >>c >d					\$> \$>	0	
<pre>\$&gt; cat a \$&gt; cat b</pre>					\$> \$>	0 0	
<pre>\$&gt; cat c \$&gt; cat d</pre>			OK		Prout \$>	0 0	
			OK	ОК	crashtest.c Docs		
					[] test.c		
A					\$>		
<pre>\$&gt; echo hola &gt; a &gt; b &gt; c \$&gt; cat a</pre>			OK		\$> \$>	0	
<pre>\$&gt; cat b \$&gt; cat c</pre>					hola \$>		
<pre>\$&gt; mkdir dir \$&gt; ls -la &gt; dir/bonjour</pre>					total 16 drwx 2 vietdu91 [] .	0	
\$> cat dir/bonjour			ОК	OK	drwxrwxewt 51 root [] []		
\$ 22					\$> Amour Tu es Horrible	0	
<pre>\$&gt; <a \$=""> cat a</a></pre>			ОК	OK .	\$>		
<pre>\$&gt; &gt;d cat <a>&gt;e \$&gt; cat a</a></pre>					\$> Amour	0 0	
<pre>\$&gt; cat d \$&gt; cat e</pre>			OK	ОК	\$> Amour	0 0	
\$> < a > b cat > hey >> d					\$> Amour Tu Es Horrible	0	
\$> cat d \$> ls			ок	ОК	\$> [] a b hey d	0	
\$> cat << hola					> \$HOME	0	
			ОК	OK .	<pre>&gt; hola /home/vietdu91</pre>		
<pre>\$&gt; cat &lt;&lt; 'hola'</pre>					\$> > \$HOME	0	
- Cac 11 Hota			ОК	OV	> hola \$HOME		
					\$>		

	<pre>\$&gt; cat &lt;&lt; "hola"</pre>	ОК	ок	> \$HOME > hola \$HOME \$>
	\$> cat << ho"la"	ок	ок	> \$HOME > hola \$HOME \$>
	\$> cat << \$HOME	ОК	ок	> prout > /home/vietdu91 > \$H0 > \$H0ME \$>
	<pre>\$&gt; cat &lt;&lt; hola &gt; bonjour \$&gt; cat bonjour</pre>	ОК	ОК	<pre>&gt; prout &gt; croutte &gt; antoine pue &gt; hola \$&gt; prout croutte antoine pue</pre> 0  6  6  7  8  9  9  9  9  9  9  9  9  9  9  9  9
E	\$> cat << hola   rev	ОК	ОК	<pre>&gt; prout &gt; croutte &gt; antoine pue &gt; hola tuorp ettuorc eup eniotna \$&gt;</pre>
HEREDOC	\$> << hola	ок	ОК	> chola > holache > hola > "hola" > 'hola' > hola
	\$> < <hola< td=""><td>ОК</td><td></td><td>&gt; 0 &gt; hola \$&gt;</td></hola<>	ОК		> 0 > hola \$>
	\$> cat <<	ок		<pre>bash: syntax error near unexpected token `newline' \$&gt;</pre>
	<pre>\$&gt; cat &lt;&lt; prout &lt;&lt; lol &lt;&lt; koala</pre>	ок	ок	> prout
	\$> prout << lol << cat << koala	ОК	ОК	<pre>&gt; prout &gt; lol &gt; cat &gt; koala bash: prout: command not found</pre> 127
	\$> << \$hola	ок		> \$hola \$>
	\$> << \$"hola"\$"b"	ок		> holab \$>
	\$> << \$"\$hola"\$\$"b"	ОК		> \$hola\$\$b \$>
	\$> << ho\$la\$"\$a"\$\$"b"	ок	ОК	> hola\$a\$\$b > ho\$la\$a\$\$nb \$>
	<pre>\$&gt; echo hola &lt;&lt;&lt; bonjour</pre>	ок		hola \$>
	\$> echo hola <<<< bonjour	ОК		bash: syntax error near unexpected token `<' 2 \$>

	<pre>\$&gt; echo hola &lt;&lt;&lt;&lt;&lt; bonjour</pre>		ОК	ок	<pre>bash: syntax error near unexpected token `&lt;&lt;' \$&gt;</pre>	2	
	\$> ls *				crashtest.c Docs Dockers Drawings Makefile minishell philosophers test.c	0	[BONUS]
		NON GERE	ОК	ок	<pre>Docs: identite.pdf loyer.pdf srcs: main.c utils.c sort.c \$&gt;</pre>		
CAR	\$> ls *.*	NON GERE	ок	ок	<pre>crashtest.c test.c \$&gt;</pre>	0	[BONUS]
VILD	\$> ls *.hola	NON GERE	ок	ОК	ls: cannot access '*.hola' : No such file or directory \$>	2	[BONUS]
>	\$> cat M*le	NON GERE	ок	ОК	[affiche le Makefile] \$>	0	[BONUS]
	\$> cat M*ee	NON GERE	ОК	ОК	cat: 'M*ee': No such file or directory	1	[BONUS]
	<pre>\$&gt; cat Make*file</pre>	NON GERE	ОК	ок	[affiche le Makefile] \$>	0	[BONUS]