

COMMAND	BASH	GAUSHELL
<b>BUILTINS SIMPLE</b>		
echo	/n	OK
echo -n	return prompt	OK
echo -nnnnn	return prompt	OK
echo This is a test	This is a test\n	OK
echo -n This is a test	This is a testarepsa@...	OK
echo -nnnn This is a test	This is a testarepsa@...	OK
echo This is a test -nnnn	This is a test -nnnn\n	OK
echo \$?	Exit error	OK
env	prints environment variables	OK
env PATH	env: 'PATH': No such file or directory	OK
cd	arepsa@c1r15s9:~\$	OK
cd .	stay in the same directory	OK
cd.	\nCommand 'cd.' not found, did you mean:...	OK
cd . ls	bash: cd: too many arguments	OK
cd ../../../../	arepsa@c1r15s9:/\$	OK
cd /usr/bin	arepsa@c1r15s9:/usr/bin\$	OK
cd nonexistent	minishell: cd: nonexistent: No such file or directory	OK
unset	return prompt	OK
unset nonexistentvariable	return prompt	OK
unset HOME	deletes HOME, returns prompt	OK
cd (after unsetting HOME)	bash: cd: HOME not set	OK
unset USER	deletes USER from env	OK
export USER=bubu (do after unset)	adds to env new user bubu	OK
export =	bash: export: `=`: not a valid identifier	OK
export ABC=\$USER	changes ABC=username	OK
export novalue	returns prompt, keeps "novalue" in export print, but does	OK
export=zzzzzz	returns prompt, doesn't add any variables	OK
export _	Export normally	OK
fdgsdfhb=	returns prompt	OK
unset blabla OLDPWD	deletes OLDPWD, returns prompt	OK
unset ALL VARIABLES	unsets all, returns prompt	OK
env (perform after unsetting all variables)	bash: env: No such file or directory	OK
pwd	prints home, returns prompt	OK
pwd args	prints home, returns prompt	OK
export	prints ordered env variables with "declare -x" before	OK
env	prints env variables	OK

[illegible]

export WWW="-aaa   "	saves variable WWW="-aaa   "	OK
export ZZ"Y=aaa"bb	exports variable ZZY="aaabb"	OK
cd ""	fica no mesmo diretorio	OK
unset MA"IL"	unsets MAIL	OK
echo 42' 42	42 42	OK
echo a'b'c'd'e'f'g'h'i'j'k'l'm'n'o'p'q'r's't'	abcdefghijklmnpqrst	OK
echo \$USER\$USER'	damachad\$USER	OK
export test="a b c"; echo \$test	a b c	OK
ls>"	No such file	OK
ls > \$nonexpansible	Ambiguous Redirect	OK
<b>EXPANDER</b>		
echo \$USER	prints username	OK
export ABC=\$USER	creates var ABC=username	OK
<b>EXIT CODES</b>		
cat nonexistent file or folder; echo \$?	exit code 1	OK
enter ./minishell; exit 300	exit code 44	OK
./not_executable_file	exit code 126	OK
< not_existing_file	exit code 1	OK
echo 1	exit code 2	OK
ls not_existing_file	exit code 2	OK
<b>PIPES</b>		
"ls"   'wc'	execute ls and wc	OK
unset USER   env   grep USER	USER=arepsa (doesn't unset variable)	OK
unset OLDPWD   ls	performs ls, doesn't unset variable	OK
echo USER   export USER=bubu	echo doesn't perform, USER is not changed, returns prompt	OK
echo Hello World     grep e	bash: syntax error near unexpected token ` '	OK
cat   cat   ls	performs ls and exits after 2 enters	OK
cat /dev/urandom   head -c 100	returns 100 random characters	OK
<b>REDIRECTS</b>		
echo hello > outfile		OK
	ls: cannot access '>': No such file or directory	OK
ls ">" ola	ls: cannot access 'ola': No such file or directory	OK
< inaccessible_file   cat -e > output	creates empty output file	OK
<	bash: syntax error near unexpected token `newline'	OK

>	bash: syntax error near unexpected token `newline'	OK
> test	creates empty output file	OK
< existingfile	Does nothing	OK
< testfile > testfile2	creates empty output file "testfile2"	OK
<b>HEREDOCS</b>		
<<	bash: syntax error near unexpected token `newline'	OK
>>	bash: syntax error near unexpected token `newline'	OK
cat << end   wc -l	count lines of heredoc file	OK
cat << end   grep e	returns heredoc lines that contain e	OK
<< end   grep e	returns prompt	OK
<< EOF cat	expands any variable inside it	OK
<< "EOF" cat	Does not expand	OK
<< 'EOF' cat	Does not expand	OK
<< "EOF' cat	unclosed quotes	OK
<b>DENTRO de HEREDOCs</b>		
\$USER e \$CASA e "\$ HOME" e '\$ HOME' "\$HOME" '\$HOME' \$HOME	(expande)USER e e "\$ HOME e '\$ HOME' "(expande)HOME" '(expande)HOME' (expande)HOME	OK
\$CASA \$ CAMA	\$ CAMA	OK
"\$CASA"   "\$ CAMA"   "\$USER"   '\$HOME' \$USER \$ "\$ HOME" << HOME	""   "\$ CAMA"   (espanse)"USER"   '(espanse)HOME' (espanse)USER \$ "\$ HOME" << HOME	OK
<b>COMMANDS</b>		
ls \$jhrghdghbd	Executes ls	OK
unset PATH; ls	bash: ls: No such file or directory	OK
/dev/urandom	bash: /dev/urandom: Permission denied	OK
\$<something that does not exist>	returns prompt	OK
<b>TO SOLVE</b>		
export test="ls -l"; \$test	runs ls -l	OK
""	Command " not found, but can be installed with:	OK
<< eof; Ctrl + C	returns prompt	OK
\$PWD	bash: <PWD>: Is a directory	OK
cat /bin/ls		OK
./bin/lsg   wc		OK
ping 8.8.8.8		OK
export test; export test=hello; echo \$test	hello	OK
""	Command " not found	OK
""   ls	executes ls, and shows "command not found"	OK

./srcs (execute existing folder)	bash: ./srcs: Is a directory	OK
a= ; echo \$?	exit code 0	OK
export us\$er=blabla		OK
unset weird*%chars		OK
cd FILENAME WITH SPACE	bash: cd: too many arguments	OK
echo hello > FILENAME WITH SPACE		OK