1. Shell enables one or more arguments to be obtained from the standard output of another command. This feature is called \_\_\_\_\_\_\_\_\_  
a) command substitution  
b) argument substitution  
c) shell substitution  
d) korn

Answer: a  
Explanation: Apart from a pipeline, shell enables connecting of two commands in another way. Shell enables one or more arguments to be obtained from the standard output of another command. This feature is called command substitution. For example, to display output like:

Today’s date is Sat Jan 05 17:25:41 IST 2002

We can use the following command

$ echo “Today’s date is `date`” // date is an argument to echo command

2. Which of the following meta-character is used in command substitution?  
a) `  
b) ‘  
c) “  
d) >

Answer: a  
Explanation: When scanning the command line, the shell looks for another metacharacter i.e. ` (the backquote) usually placed on the top-left of our keyboard, and it should not be confused with a single quote (‘).

3. Command substitution is enabled in single quotes also.  
a) True  
b) False

Answer: b  
Explanation: Command substitution is enabled only when we use double quotes. If we use single quotes, it will not work. For example,

$ echo ‘today’s date is `date` ‘

Output will be: today’s date is `date

$ echo “today’s date is `date` “

Output will be: today’s date is Sat Jan 05 17:25:41 IST 2017

4. Which symbol is used for assigning a value to variables?  
a) $  
b) &  
c) =  
d) @

Answer: c  
Explanation: The shell supports variables that are useful both in the command line and shell scripts. These variables are called shell variables like pwd. A variable assignment is of the form variable=value(no spaces around =). For example,

$ count=10 // a variable named count is assigned a value of 10

5. Which symbol is used for evaluation of variables?  
a) $  
b) &  
c) =  
d) @

Answer: a  
Explanation: The shell supports variables that are useful both in the command line and shell scripts. These variables are called shell variables like TERM and SHELL. A variable assignment is of the form variable=value(no spaces around =), but its evaluation requires the $ as a prefix to the variable name. For example,

$ count=10

$ echo $count // output will be 10

6. Which of the following is a correct initialization of variables to null strings?  
a) x=  
b) x=’ ‘  
c) x=” “  
d) x=, x=’ ‘, x=” “

Answer: d  
Explanation: All shell variables are initialized to null strings by a string. While explicit assignment of null strings can be performed with x=’ ‘ or x=” “ or x=

7. A variable can be removed using \_\_\_\_\_  
a) unset  
b) readonly  
c) del  
d) bash

Answer: a  
Explanation: A variable can be removed using the unset command. unset in an internal command. For example, if we want to undefine a variable x then,

$ unset x

8. readonly command is used to protect a variable from reassignment.  
a) True  
b) False

Answer: a  
Explanation: A variable can be protected from reassignment by readonly command. unset is also an internal command. For example, to protect a variable x from reassignment use the following command,

$ readonly x // x can’t be reassigned now

9. What will the result when we evaluate this statement?

$ directory=’pwd’=`pwd`

a) output of pwd command along with string pwd=  
b) undefined output  
c) erroneous  
d) directory variable will hold string pwd

Answer: a  
Explanation: In the above statement, the string pwd, = and the output of pwd command are concatenated and saved in the directory variable. So the above statement will evaluate the current working directory as:

pwd=/home/usr10/doc

10. Which of the following is not a system defined variable?  
a) $PATH  
b) $HOME  
c) $SHELL  
d) $cd

Answer: d  
Explanation: PATH is the shell variable which stores the list of directories that the shell searches for locating commands. HOME prints our home directory while SHELL prints the absolute pathname of our login shell. cd is a command which is used for changing directories or moving through file hierarchy.

11. Which of the following is an invalid variable?  
a) \_user  
b) us01  
c) -txtfile  
d) txt123

Answer: c  
Explanation: UNIX restricts some rule for defining a variable. A variable name must begin with either an underscore (\_) or alphanumeric character followed by one or more alphanumeric or underscore characters.

Following are some valid variable names:

User01 user\_01 \_user67

Following are some invalid variable names:

2\_var user! -textfile

12. The command is valid.

$ ls -lRa $HOME > home.ls

a) True  
b) False

Answer: a  
Explanation: Above command will save the entire home directory structure including the hidden files in a file named home.ls because we have used -l, -a, -R, all the files including hidden ones are recursively saved in a separate file named home.