

**1. \*Ce este un script shell?\***

- A) Un program compilat
- B) Un set de comenzi scrise pentru a fi executate de un shell
- C) O aplicație grafică
- D) Un sistem de operare

**2. \*Care este sintaxa corectă pentru a crea o variabilă în shell?\***

- A) var = value
- B) var=value
- C) var : value
- D) var -> value

**3. \*Cum se exportă o variabilă de mediu?\***

- A) set var=value
- B) export var=value
- C) declare var=value
- D) define var=value

**4. \*Ce face comanda echo \$HOME?\***

- A) Afișează numele utilizatorului
- B) Afișează directorul personal al utilizatorului
- C) Afișează variabilele de mediu
- D) Afișează calea scriptului curent

**5. \*Ce operator se folosește pentru a verifica dacă un fișier există?\***

- A) -e
- B) -f
- C) -d
- D) -x

**6. \*Care este semnificația comenzii #!/bin/bash?\***

- A) Indică tipul de fișier
- B) Specifică interpretorul care va rula scriptul
- C) Este un comentariu
- D) Nu are semnificație

**7. \*Ce se întâmplă dacă folosești ghilimele simple în shell?\***

- A) Variabilele sunt expandate
- B) Totul este interpretat literal
- C) Comenzile sunt executate
- D) Se generează o eroare

**8. \*Cum se definește o funcție în shell?\***

- A) function\_name() { commands; }
- B) def function\_name { commands; }
- C) func function\_name() { commands; }
- D) function function\_name() { commands; }

**9. \*Ce face comanda shift în scripturile shell?\***

- A) Schimbă poziția argumentelor
- B) Creează o variabilă
- C) Verifică existența unui fișier
- D) Afișează argumentele primite

**10. \*Care este rezultatul comenzii echo \$(( 5 + 3 ))?\***

- A) 5 + 3
- B) 8
- C) 53
- D) 0

**11. \*Ce operator este folosit pentru a verifica dacă două stringuri sunt diferite?\***

- A) =!
- B) !=
- C) <>
- D) -ne

**12. \*Cum se citește o linie de input de la utilizator?\***

- A) read input
- B) input = read
- C) input < read
- D) scan input

**13. \*Ce face comanda test -d /path/to/directory?\***

- A) Verifică dacă directorul este gol
- B) Verifică dacă calea este un director
- C) Verifică dacă calea este un fișier
- D) Verifică dacă directorul există și este accesibil

**14. \*Cum se utilizează un bloc if în shell?\***

- A) if [ condition ]; then commands; fi
- B) if (condition) { commands; }
- C) if condition then commands
- D) if { condition } commands

**15. \*Ce rol are comanda && în shell?\***

- A) Înlănțuie două comenzi
- B) Execută a doua comandă doar dacă prima a eșuat
- C) Execută a doua comandă doar dacă prima a avut succes
- D) Afișează rezultatul unei comenzi

**16. \*\*Care este diferența între \$\* și \$@?\***

- A) Niciuna
- B) \$\* este o listă de argumente; \$@ este un singur string
- C) \$\* este un singur string; \$@ este o listă de argumente
- D) \$\* are variabile globale; \$@ are variabile locale

**17. \*Ce face comanda for var in list; do commands; done?\***

- A) Execută comenzi pentru fiecare element din listă
- B) Verifică condiții
- C) Afișează lista
- D) Permite utilizarea variabilelor

**18. \*Ce face comanda until?\***

- A) Execută comenzi până când o condiție devine adevărată
- B) Execută comenzi atâta timp cât o condiție este falsă
- C) Afișează o condiție
- D) Oprește execuția scriptului

**19. \*Cum se verifică numărul de argumente primite de un script?\***

- A) \$#
- B) \$\*
- C) \$@
- D) \$0

**20. \*Ce va afișa comanda echo "\$USER"?\***

- A) Numele scriptului
- B) Numele utilizatorului curent
- C) Valoarea variabilei USER
- D) Numele shell-ului

**21. \*Cum se declară o variabilă locală?\***

- A) local var=value
- B) var=value
- C) declare var=value
- D) set var=value

**22. \*Ce face comanda break într-un ciclu?\***

- A) Oprește execuția scriptului
- B) Iese din ciclu
- C) Continuă ciclul
- D) Afișează un mesaj

**23. \*Când se folosește fi?\***

- A) La sfârșitul unui bloc if
- B) La începutul unui script
- C) Într-un bloc case
- D) Niciodată

**24. \*Ce se întâmplă dacă folosești ghilimele duble în shell?\***

- A) Totul este interpretat literal
- B) Variabilele sunt expandate
- C) Se generează o eroare
- D) Comenzile sunt executate direct

**25. \*Cum se verifică dacă un fișier este executabil?\***

- A) -e
- B) -f
- C) -x
- D) -d

**26. \*Ce face comanda while?\***

- A) Execută comenzi atâta timp cât o condiție este adevărată
- B) Execută comenzi o singură dată
- C) Afișează un mesaj
- D) Iese din script

**27. \*Cum se utilizează un bloc case?\***

- A) case variable in { commands; }
- B) case variable in
- C) case variable; do commands; esac
- D) case (variable) { commands; }

**28. \*Ce face comanda continue într-un ciclu?\***

- A) Oprește execuția scriptului
- B) Iese din ciclu
- C) Sare la următoarea iterație a ciclului
- D) Afișează un mesaj

**29. \*Cum se redirecționează ieșirea standard într-un fișier?\***

- A) command > file
- B) command < file
- C) command >> file
- D) command | file

**30. \*Ce operator se folosește pentru a verifica dacă un număr este mai mare decât altul?\***

- A) -gt
- B) >
- C) >
- D) -lt

**31. \*Ce va returna comanda [ -f /etc/passwd ]?\***

- A) Adevărat dacă /etc/passwd este un fișier
- B) Fals dacă /etc/passwd nu există
- C) Adevărat dacă /etc/passwd este un director
- D) Niciuna dintre cele de mai sus

**32. \*Cum se utilizează trap în shell?\***

- A) Pentru a captura semnalele
- B) Pentru a opri execuția scriptului
- C) Pentru a redirecționa ieșirea
- D) Pentru a defini o funcție

**33. \*Ce face comanda let?\***

- A) Evaluează expresii aritmetice
- B) Definirea variabilelor
- C) Verifică condiții
- D) Afișează rezultatul

**34. \*Ce reprezintă \$? în shell?\***

- A) Ultimul argument
- B) Ultimul cod de ieșire al comenzii
- C) Toate argumentele
- D) Niciuna dintre cele de mai sus

**35. \*Cum se scrie un comentariu într-un script shell?\***

- A) // comment
- B) # comment
- C) /\* comment \*/
- D) -- comment

**36. \*Ce se întâmplă când se folosește exit într-un script?\***

- A) Scriptul continuă
- B) Scriptul se oprește și returnează un cod de ieșire
- C) Scriptul este salvat
- D) Niciuna dintre cele de mai sus

**37. \*Care este semnificația comenzii source filename?\***

- A) Execută un script într-un subshell
- B) Încarcă un script în shell-ul curent
- C) Salvează un script
- D) Afișează un script

**38. \*Cum se utilizează operatorul || în shell?\***

- A) Execută a doua comandă dacă prima a avut succes
- B) Execută a doua comandă dacă prima a eșuat
- C) Înlănțuie două comenzi
- D) Niciuna dintre cele de mai sus

**39. \*Ce face comanda pwd?\***

- A) Afișează numele utilizatorului
- B) Afișează calea directorului curent
- C) Schimbă directorul curent
- D) Niciuna dintre cele de mai sus

**40. \*Cum se verifică dacă un script a fost executat cu succes?\***

- A) if [ \$? -eq 0 ]; then
- B) if [ \$? -ne 0 ]; then
- C) if [ \$0 -eq 0 ]; then
- D) if [ \$0 -ne 0 ]; then

**41. \*Care este funcția comenzii basename?\***

- A) Afișează numele complet al unui fișier
- B) Returnează numele fișierului fără căile
- C) Afișează calea unui fișier
- D) Niciuna dintre cele de mai sus

**42. \*Ce face comanda cut?\***

- A) Îmbină linii de text
- B) Extrage secțiuni din fiecare linie
- C) Afișează conținutul fișierului
- D) Niciuna dintre cele de mai sus

**43. \*Cum se definește un array în shell?\***

- A) array=(value1 value2 value3)
- B) array{value1, value2, value3}
- C) array[value1, value2, value3]
- D) array<value1, value2, value3>

**44. \*Ce face comanda grep?\***

- A) Afișează conținutul unui fișier
- B) Caută un pattern în text
- C) Schimbă numele fișierului
- D) Niciuna dintre cele de mai sus

**45. \*Cum se combină mai multe comenzi într-un singur script?\***

- A) Fiecare comandă pe o linie nouă
- B) Folosind ; între comenzi
- C) Folosind | între comenzi
- D) Toate cele de mai sus

**46. \*Ce face comanda chmod?\***

- A) Schimbă proprietarul fișierului
- B) Schimbă permisiunile unui fișier
- C) Afișează conținutul fișierului
- D) Niciuna dintre cele de mai sus

**47. \*Cum se verifică dacă un script are permisiuni de execuție?\***

- A) if [ -x script.sh ]; then
- B) if [ -e script.sh ]; then
- C) if [ -f script.sh ]; then
- D) Niciuna dintre cele de mai sus

**48. \*Ce face comanda find?\***

- A) Caută fișiere și directoare
- B) Afișează conținutul unui fișier
- C) Schimbă numele unui fișier
- D) Niciuna dintre cele de mai sus

**49. \*Cum se obține numărul de elemente dintr-un array în shell?\***

- A) \${#array}
- B) array.count
- C) count(array)
- D) size(array)

**50. \*Ce face comanda tar?\***

- A) Comprimă fișiere
- B) Dezarhivează fișiere
- C) Crează arhive
- D) Toate cele de mai sus

1. B
2. B
3. B
4. B
5. A
6. B
7. B
8. A
9. A
10. B
11. B
12. A
13. B
14. A
15. C
16. C
17. A
18. A
19. A
20. B
21. A
22. B
23. A
24. B
25. C
26. A
27. B
28. C
29. A
30. A
31. A
32. A
33. A
34. B
35. B
36. B
37. B
38. B
39. B
40. A
41. B

42. B
43. A
44. B
45. D
46. B
47. A
48. A
49. A
50. D