

1. kérdés

1 / 1 pont

Mire használtak a speciális fájlnev karakterek (*,?,[],!)?

☐ Nincsenek ilyenek, csak a reguláris kifejezés karakterei használtak.



Egy speciális kifejezés formával tudunk egy vagy több névre hivatkozni vagy megadni.



A szöveg összehasonlításnál ismert módon tudunk relációkat megadni ezekkel speciális karakterekkel.



A shell az aposztrof vagy idézőjel karaktereket használja a megadott karakterek mellett automatikusan.

2. kérdés

1 / 1 pont

Lehet-e karakter kódtáblával adatokat, szöveget titkosítani?

☐ Nem, a kódtábla szövegek megjelenítésére való csak!



A filmekben is gyakran használt módszer, akár használható is!



Mindíg ezt használjuk, vennünk kell az RSA kódtáblát, ami alapján tudunk titkosítani!



Csak az UTF-16 és az kódtábla nem használható erre, minden más igen!

Helytelen

3. kérdés

0 / 1 pont

Mondjon példát arra, hol használhat reguláris kifejezéseket?

☐

Bármely szövegrészben használhatunk reguláris kifejezéseket. Az aktuális parancs fogja értelmezni a speciális karaktereket.

☐

Csak a szűrő parancsok szeretik, ha más parancsnál használjuk hibajelzést kapunk!

☐

Bármely szövegrészben használhatunk reguláris kifejezéseket. Más kérdés ki mit csinál vele!

☒

Bármely szövegrészben használhatunk reguláris kifejezéseket. Azokat mindig az extended reguláris kifejezés szabályai szerint kell értelmezni.

Helytelen

4. kérdés

0 / 1 pont

A mai 64 bites CPU egységek segítségével tudunk-e 64 bitnél nagyobb méretű műveleteket végrehajtani? (Például tudunk-e 139 bites műveleteket használni?)

☐

Igen, a CPU bitmérete gyakorlatilag semmit nem befolyásol, legfeljebb nem sokat segít.

☒

Igen lehet, de csak adott méretekben, 32 bitenként növelve.

☐

Nem tudunk, mert erre a CPU képtelen, hiszen csak 64 bites.

☐

Igen lehet, de csak adott méretekben, 16 bitenként növelve.

5. kérdés

1 / 1 pont

A mai számítógép architektúrák világában létezhet-e olyan számítógép ami csak memóriát tartalmaz a CPU mellett?

- ☐ Nem, ilyen nem lehetséges.
- ☐ Igen, megengedett, és csak a Begin-Process-End blokkok használható az ilyen felépítés mellett.
- ☒ Persze, elvileg igen, gyakorlatilag nincs ilyen.
- ☐ Nem, nem, ma csak logikai áramkörök találhatók a számítógépbe, így nem lehet ilyen.
- ☐ Nem lehetséges ez, mert a CPU eleve tartalmaz memóriát, így nem kell még egy! Jóból is megárt a sokk!

6. kérdés

1 / 1 pont

Mire szolgál a "here input" funkció Shell scriptben?

☐

A Unix rendszerbeli összes speciális karaktert itt nem használhatjuk mikor itt megadjuk a bemenetet!

☐

Nincs a shell scriptben ilyen lehetőség.

☐

A z Input-Here Unix parancs is adja ezt a lehetőséget.

☒

Ez több soros szöveg definiálására ad hasonló lehetőséget, ami az adott parancs bemenetére kerül.

telen

7. kérdés

0 / 1 pont

Milyen hátultesztelő ciklust ismer shell scriptben?

☒

Csak az Until ciklus tesztel hátul.

☐

Nincs ilyen "fenevad".

☐

A do...done while feltétel a hátultesztelő ciklus.

☐

Minden ciklus hátul is tesztel! Ahova csak írjuk.

Helytelen

8. kérdés

0 / 1 pont

A lebegőpontos számábrázolás karakterisztikája mit határoz meg?

- ☐ A szám nagyságrendjét.
- ☒ A szám pontosságát.
- ☐ A szám előjelét.
- ☐ Ez a mantisszával felcserélhető érték, így attól függ fel van-e cserélve!

Helytelen

9. kérdés

0 / 1 pont

A test utasítást miért részesítjük előnyben a logikai kifejezések kiértékelésekor az expr utasítással szemben?

- ☐ Egyáltalán nem létesítjük előnyben! Sőt nagyon is szeretjük.
- ☐ Fordítva, az expr utasítás többet tud, például logikai és műveletet is támogat.
- ☐ Azért, mert több lehetőséget tartalmaz. Az expr csak egy "gyenge utánszat".
- ☒ Azért mert nem nyújt alap aritmetikai műveleteket. (+,-,*,/,%)

10. kérdés

1 / 1 pont

Melyik operációs rendszerben nem használhatók az ACL (Access Control List) alapú hozzáférési listák?

- ☒ Ez nem operációs rendszer, hanem fájlrendszer függő.
- ☐ A SUSE Enterprise Real Time világban, mert ott mindenkinek van hozzáférési joga.
- ☐ Csak a MacOS elődverzióiban (OSX alatt) nem használhatók.
- ☐ A ma már elavult 32 bites operációs rendszerekben.