

1. kérdés

1 / 1 pont

Mit nevezünk szekvenciális inputfájlnak?

☐

Ez egy régi adattárolási technológia, amely úgy helyezi el adatok sorozatát egy háttértárolón (jellemzően egy mágnesszalagon), hogy azokat csak sorban, elejétől hátrafelé haladva, egyesével tudjuk visszakeresni.

☐

Bármilyen gyűjtemény lehet szekvenciális inputfájl, ha rendelkezik olyan "olvasás" művelettel, amelyik egy elolvasott elemet rögtön ki is vesz a gyűjteményből.

☒

Az olyan azonos típusú elemek sorozatát, amelyre egy műveletet engedünk csak meg: a sorozat első elemének kiolvasását (és egyben kitörlését).

☐

Az olvasásra megnyitott szöveges állományokat.

2. kérdés

1 / 1 pont

Az alábbiak közül mely állítások igazak a szekvenciális inputfájl read műveletére?

☐

Ha a fájl aktuális elemét jelző mutató nem a fájl vége utánra mutat, akkor, kiolvassa az aktuális elemet, és a mutatót eggyel tovább lépteti.

☐

Üres fájl esetén abortál (hibát okoz).

☒

Ha a fájl nem üres, akkor a read művelet leválasztja az első elemét a fájlnak.

☒

Üres fájl esetén jelzi, hogy az olvasás sikertelen.

3. kérdés

1 / 1 pont

Szabad-e a szekvenciális inputfájl read műveletét közvetlenül egymás után, minden egyéb ellenőrzés nélkül, kétszer végrehajtani?

- ☐ Nem, mert minden olvasás előtt meg kell győződnünk arról, hogy a fájl nem üres.
- ☐ Csak akkor, ha az első olvasás előtt meggyőződünk arról, hogy a fájl nem üres.
- ☐ Csak akkor, ha feltételezhetjük, hogy az inputfájl kellő számú elemet tartalmaz.
- ☒ Igen, mert ha a fájl ki is ürülne, egy újabb read művelet az üres fájlra is értelmezett.

4. kérdés

1 / 1 pont

Hogyan valósítható meg egy szekvenciális inputfájl (**file**) elemeinek (**elem**) felsorolása?



A First() műveletet az **status,elem,file:read** implementálja, a Next() műveletet az **if status=norm then status,elem,file:read**, a Current() művelet az **elem** értékét adja vissza, az End() művelet pedig a **status=abnorm** logikai értéket.



A First() és a Next() műveletet az **status,elem,file:read** implementálja, a Current() művelet az **elem** értékét adja vissza, az End() művelet pedig a **status=abnorm** logikai értéket.



Ez attól függ, hogy a fájl egy eleme elemi típusú-e (például karakter, vagy egész szám), vagy összetett.



A First(), a Next(), és az End() műveletet a **status,elem,file:read** művelet váltja ki, a Current() művelet az **elem** értékét adja vissza.

5. kérdés

1 / 1 pont

Mi az az előre olvasási technika?

☐ Mielőtt olvas egy szekvenciális inputfájlból, meggyőződik arról, hogy nem üres.

☐

Amelyik egy háttértárolón elhelyezett adatok teljes sorozatát, vagy annak egy-egy blokkját előzetesen átmásolja a memóriába, hogy majd onnan egyesével olvashassa az elemeket.

☐

Szekvenciális inputfájl elemeinek feldolgozásánál az elemeket sorban egymás után, elejétől a végéig haladva olvassa ki.

☒ Először olvas, majd megvizsgálja, hogy sikerült-e így újabb elemhez hozzájutni.

6. kérdés

1 / 1 pont

Milyen ciklus-szervezést igényel az előre olvasási technika?

☐ Számlálós ciklusra van szükség.

☐

Hátul-tesztelő ciklust kell alkalmazni, hogy az előre olvasást a ciklusmag elejére helyezhessük el.

☒

A ciklus előtt, illetve a ciklusmag végén kell a ciklusmag következő végrehajtásához szükséges elemet beolvasni a fájlból.

☐

A ciklus feltételnek ellenőriznie kell, hogy a fájl még nem üres, mielőtt a ciklusmag elején beolvasnánk a fájl következő elemét.

7. kérdés

1 / 1 pont

Mit nevezünk szekvenciális outputfájlnak?

☐ Az írásra megnyitott szöveges állományokat lehet így nevezni.



Bármilyen gyűjtemény lehet szekvenciális outputfájl, ha rendelkezik olyan "írás" művelettel, amelyik egy elemet tesz be a gyűjteménybe.



Az olyan azonos típusú elemek sorozatát (speciális gyűjtemény), amely kezdetben üres, és csak a sorozat végéhez történő hozzáírást engedjük meg.



Ez egy régi adattárolási technológia, amely egy háttértárolón (jellemzően egy mágnesszalagon) helyezi el adatok sorozatát sorban egyesével az elejétől hátrafelé haladva.

8. kérdés

1 / 1 pont

Az alábbiak közül mely állítások igazak a szekvenciális outputfájlra?



Az "x : write(e)" szimbólummal jelölt művelete egyenértékű az "x := x \oplus <e>" értékadással.



Ez egy olyan sorozat, amelynek végéhez újabb elemeket lehet illeszteni.



Üres fájl esetén az írás abortál (hibát okoz).



Két művelete van: kezdetben az üres sorozat létrehozása, és a sorozat végéhez történő hozzáírás.

9. kérdés

1 / 1 pont

A tanult hat algoritmus minta melyikével lehet megoldani az olyan fájlfeldolgozási feladatokat, mint a másolás, a kiválogatás, a szétválogatás.



Ezeket nem lehet a tanult hat algoritmus minta egyikére sem visszavezetni.



Ezek mind az összegzés mintára vezethetők vissza.



Ezek maguk is a tanult algoritmus minták közé tartoznak.



A másolásokat az összegzésre, a kiválogatásokat a kiválasztás programozási tételére, a szétválogatást több kiválasztás algoritmusra vezethetjük vissza.

10. kérdés

1 / 1 pont

Szükséges-e valamilyen módosítást végezni a lineáris keresés algoritmusán, ha a felsorolóját egy szekvenciális inputfájl felsorolása adja.



Ez az algoritmus minta egyáltalán nem alkalmazható szekvenciális inputfájl feldolgozására.



Nem, mivel a felsoroló műveletek egyértelműen kiválthatók a read művelettel.



Igen, a ciklus előtt szükség van egy előre olvasásra.



Igen, a ciklusmagba egy elágazást kell beletenni, amely vizsgálja, hogy a fájl üres-e.