

PTI Záróvizsga tételsor 2022 június

A tétel mellé húzni kell egy egyszerű feladatot, melyre programot kell írni valamely tanult programozási nyelven (papíron). A feladatokat lsd. lentebb. Minden felelet tehát 3 részből tevődik össze:

- a. Szakdolgozat védeése
- b. Az „egyszerű feladat”
- c. A kihúzott tétel

A szakdolgozat védeésénél válaszolni kell tudni a bíráló által feltett kérdés(ek)re, melyeket mindenki megismerhet a vizsga előtt egy héttel. Amennyiben kiderül, hogy plagizált a dolgozat, vagy nem a jelölt írta, elégtelen a szakdolgozatvédeés érdemjegye (s így a záróvizsgáé is).

A felelet során az „egyszerű feladat”-ot tökéletesen kell megoldani, eltekintve a papíron való programozás miatt előforduló egy-két szintaktikai hibától, - azaz szemantikailag tökéletes feladatokat kis szintaktikai hibával még elfogadunk -, ellenben elégtelen a záróvizsga érdemjegye. A feladatokat az alábbi programozási nyelvek valamelyikén kell megírni: C, Java, C#, ezekből a jelölt **szabadon** választhat. Tilos olyan függvényt/metódust használni, mely közvetlenül a megoldást adja, pl. az 1. feladatnál *isPrime()*. Hasonlóan nem megengedett rendező és kereső metódusok/függvények használata.

Ha a felelet során a kihúzott tételre vonatkozó feleletet elégtelennek ítéli a bizottság, akkor elégtelen a záróvizsga érdemjegye.

Esetleges kérdéseikkel keressenek bátran a falucskai.janos@nye.hu e-mail címen.

Tételsor

- 1. A C programozási nyelv I.:** Adattípusok, deklarációik, feltételes utasítások.
Általános ismeretek: adat és információ, entrópia fajtái, kifejezések infix és postfix alakja.
Mesterséges intelligencia: Keresési problémák állapottér-reprezentációja, példák.
Neminformált keresési eljárások (mélységi, szélességi, optimális).
- 2. A C programozási nyelv II.:** Ciklusszervezési lehetőségek, függvénykezelés, paraméterkiértékelés, hatáskörkezelés (statikus, dinamikus).
Általános ismeretek: számrendszerek, számábrázolás (fix és lebegőpontos), karakter, szöveg és logikai adat ábrázolása.
Mesterséges intelligencia: A heurisztika fogalma, példák. A* algoritmus. Az A* algoritmus teljessége. Kétszemélyes, teljes információjú, determinisztikus játékok: a stratégia fogalma, minimax-algoritmus, alfa-béta vágás.
- 3. A C programozási nyelv III.:** Tömbök, mutatók, dinamikus memóriakeresés. Karakterlánc-kezelés.
Formális nyelvek és automaták: környezetfüggetlen nyelvtanok, CNF, CYK algoritmus.
Hálózatok: adatkapcsolati protokollok, rétegek. Lokális hálózatok. Az internet alapjai, HTML.
- 4. Adatszerkezetek és algoritmusok:** Funkcionális specifikáció. Programozási tételek: keresés, rendezés, döntés, kiválogatás.
Adatbázis-rendszerek: A relációs adatmodell. Egyed, attribútum, reláció és kapcsolat. Kulcs, idegen kulcs, hivatkozási integritás. Kényszerfeltételek az adatbázis elemein. Triggerek.
Számításelmélet: Turing-gépek, Church-tézis, megállási probléma, algoritmikusan eldönthetetlen problémák. Logikai függvények megadása, KNF, DNF, logikai hálózatok. Tár- és időbonyolultság.
- 5. Adatszerkezetek és algoritmusok:** sor, verem, hiányos mátrix, láncolt lista, bináris fák műveleteinek algoritmusai.
Operációs rendszerek: Folyamatkezelés és -ütemezés. Memóriakezelés. Állománykezelés.
Formális nyelvek és automaták: Az üresszó-lemma. Véges automata fogalma, fajtái, véges automaták determinisztikussá tétele.
- 6. Objektum-orientált programozás:** OOP, típusok és konverzióik, operátorok, utasítások. Metódusok, osztálykészítés, láthatóság, konstruktor.
Formális nyelvek és automaták: Ábécé, szó, nyelv, nyelvtan fogalma. Chomsky-féle nyelvtani osztályok és az általuk generált nyelvosztályok tartalmazási hierarchiája.
Számítógép-architektúrák: logikai áramkörök, kombinációs logikai hálózatok (fél és teljes összeadó, multiplexer, demultiplexer, dekóder).
- 7. Objektum-orientált programozás:** Öröklődés, túlterhelés, polimorfizmus. Kivételkezelés.
Adatbázis-rendszerek: Nézetablák relációs adatbáziskezelőkben. Indexelés a táblákon – mikor használjuk? Az adatbázis-tervezés elmélete: funkcionális függőségek és normalizáció – Boyce-Codd normálforma (BCNF). Anomáliák nem normalizált adatbázissémák esetén. Az E/K modell és átfordítása relációs adatmodellé.
Programozási technológiák: Nyomkövetés és hibakeresés, egységtesztelés, naplózás. Kollekciónak használata, relációs adatbázisok kezelése OO programozási nyelvekben.

8. **Objektum-orientált programozás:** Osztály- és példány inicializálás, konstruktor. Interfészek. Generikus programozás, összetett adatszerkezeteket implementáló osztályok és fontosabb műveleteik.
A rendszerfejlesztés technológiája: rendszerfejlesztési modellek, tervezés, tesztelés, UML osztálydiagram. Verziókezelés.
Számítógép-architektúrák: A mikroelektronika alapjai (félvezetők, dióda, tranzistorok, fajtái és az általuk megvalósítható kapuk). A CPU és felépítése. Integrált áramkörök. Memóriák fajtái, csoportosításuk.
9. **SQL:** Adatdeklarációs résznyelv (DDL), a CREATE TABLE és ALTER TABLE utasítás lehetőségei. Adatlekérdező nyelv (SELECT): rendezés, szűrés, csoportosítás, többtáblás lekérdezések, az INNER JOIN és OUTER JOIN különbsége. Adatmódosító (DML) résznyelv: INSERT, UPDATE, DELETE. Beágyazott allekérdezések lehetőségei: IN, EXISTS, ALL, ANY. Kapcsolt allekérdezés.
Programozási technológiák: Tervezési minták egy OO programozási nyelvben. MVC, mint modell-nézet-vezérlő minta és néhány másik tervezési minta.
Hálózatok: Topológiák és architektúrák. Az OSI modell. Fizikai átviteli jellemzők és módszerek, közeg hozzáférési módszerek.

Feladatsor

1. Írjon olyan függvényt vagy metódust, amely egy természetes számról eldönti, hogy prímszám-e, vagy sem!
2. Írjon olyan függvényt vagy metódust, amely egy természetes számról eldönti, hogy tökéletes szám-e, vagy sem! (pozitív osztóinak összege a szám kétszerese)
3. Írjon olyan függvényt vagy metódust, amely egy karakterláncban vagy sztringben véletlenszerűen összekeveri a karaktereket (véletlenszám-generátor használható)!
4. A következő közelítő formulát használva írjon függvényt vagy metódust, amely egy valós szám négyzetgyökét adja vissza! Használja az $x_{k+1} = 1/2 * (x_k + a/x_k)$ sorozatot, amely a négyzetgyökéhez konvergál, ha $x_1 = 1$.
5. Írjon függvényt vagy metódust, amely egy valós szám köbgyökét adja vissza! Használja az $x_{k+1} = 1/3 * (2 * x_k + a/x_k^2)$ sorozatot, amely a köbgyökéhez konvergál, ha $x_1 = 1$.
6. Írjon függvényt vagy metódust, amely kiszámolja az n -edik Fibonacci számot! A Fibonacci sorozatot az $a_n = a_{n-2} + a_{n-1}$ rekurzióval definiálja ($n > 2$), ahol $a_1 = a_2 = 1$.
7. Írjon olyan függvényt vagy metódust, amely egy természetes számhoz visszaadja azt a legnagyobb egész kitevős hatványát, amely még éppen kisebb, mint 567!
8. Írjon olyan függvényt vagy metódust, amely egy természetes szám esetén kiírja, hogy a 9-es számjegyből hány darabot tartalmaz (ne alakítsa át sztringgé/karaktertömbbé)!
9. Írjon olyan függvényt vagy metódust, amely egy természetes számról eldönti, hogy a kettes számrendszerbeli felírásában a jobbról második bitje 1 vagy 0 (ne alakítsa át sztringgé/karaktertömbbé)!
10. Írjon olyan függvényt vagy metódust, amelynek paramétere egy $1 < x < 10$ természetes szám, és kiírja az 1,3,4,6,7,9,10,12,... sorozat első öt x -szel osztható elemét, azaz a sorozat $i+1$ -edik tagja 2-vel nagyobb az i -ediknél, ha i páratlan, s eggyel nagyobb az i -ediknél, ha i páros!
11. Írjon olyan függvényt vagy metódust, amely a paraméterében megadott természetes szám pozitív osztóinak számával tér vissza!
12. Írjon olyan függvényt vagy metódust, amely egy karakterláncból vagy sztringből a számjegyek kivételével minden karaktert eltávolít!
13. Írjon olyan függvényt vagy metódust, amely egy karakterláncról vagy sztringről eldönti, hogy palindróma-e! (Balról olvasva ugyanaz, mint jobbról olvasva.)
14. Írjon olyan függvényt vagy metódust, amely egy, az angol ábécé betűit tartalmazó karakterláncban vagy sztringben minden szó kezdőbetűjét nagybetűre alakítja!
15. Írjon olyan függvényt vagy metódust, amely egy karakterláncból vagy sztringből eltávolítja egy megadott karakter összes előfordulását!
16. Írjon olyan függvényt vagy metódust, amely megszámolja egy adott karakterlánc vagy sztring összes előfordulását egy másik karakterláncban vagy sztringben!
17. Írjon olyan függvényt vagy metódust, amely kiírja az angol kisbetűs ábécé azon betűit,

melyek ASCII kódja négyzetszám!

18. Írjon olyan függvényt vagy metódust, amely előállít egy 5 karakterből (angol kisbetűs ábécé karaktereit használva) álló véletlen karakterláncot vagy sztringet! Biztosítsa, hogy minden 5 hosszú különböző betűkből álló sztring egyenlő valószínűséggel kerüljön kiválasztásra, feltéve, hogy a választott programozási nyelv véletlenszám-generátora egyenletes eloszlást biztosít!
19. Írjon olyan függvényt vagy metódust, amely egy karakterláncba vagy sztringbe beszúr egy „a” karaktert véletlenül választott pozícióba (véletlenszám-generátor használható)!
20. Adjon olyan függvényt vagy metódust, ami adott két pozitív egész paramétere esetén megadja $(n \text{ alatt a } k) = n! / k!(n-k)!$ értékét. Használjon rekurziót!
21. Adjon olyan metódust vagy függvényt, ami eldönti, hogy a paramétereként megadott (pozitív egészekből álló) nemüres tömbben van-e olyan szám, ami az összes többi osztja. (Maradékszámító függvény használható).
22. Adjon olyan metódust vagy függvényt, ami eldönti, hogy a paramétereként megadott (pozitív egészekből álló) nemüres tömbben van-e olyan szám, ami az összes többinél többször fordul elő.
23. Adjon olyan metódust vagy függvényt, ami visszaadja, hogy a paramétereként megadott (pozitív egészekből álló) nemüres tömbben melyik index az, ahol a leghosszabb folyamatosan növekvő részsorozat kezdődik. Ha több ilyen index is van, az utolsót adja vissza.
24. Adjon olyan metódust vagy függvényt, ami visszaadja, hogy a paramétereként megadott (pozitív egészekből álló) nemüres tömbben melyik az a legkisebb index, amire az index előtti elemek összege meghaladja a tömb első két elemének szorzatát. Ha nincs ilyen, 0-t adjon vissza.
25. Adjon egy metódust vagy függvényt, ami paraméterként adott *s* sztring/karaktertömb, *c* karakter és *n* pozitív egész szám esetén megadja, hogy a *c* karakter *n*-edik előfordulása hányadik pozíción van az „*s*” sztringben.
26. Adjon metódust vagy függvényt, ami a paraméterként kapott, egészekből álló rendezett tömbben a tömb hosszának logaritmusával arányos lépésszám alatt megkeresi a paraméterként kapott egész első előfordulásának indexét, illetve ha nincs ilyen, akkor -1-et ad vissza. (pl. a bináris keresés)
27. Írjon függvényt vagy metódust, mely visszaadja két egész paramétere szorzatának balról második számjegyét! (a megoldás során ne használjon sztringeket/karaktertömböket)
28. Írjon függvényt vagy metódust, mely eldönti, hogy a paraméterként kapott 5x5-ös /karakterekből álló/ tömbben a főátlóban van-e olyan elem, mely a főátlón kívül is megjelenik a tömbben!
29. Írjon függvényt vagy metódust, mely valós típusú paraméterének azt a számjegyét adja vissza, amelyik a tizedes pont után áll! (a megoldás során ne használjon sztringeket/karaktertömböket)
30. Írjon függvényt vagy metódust, mely pozitív egész paraméterét fordítva adja vissza, pl. fordít(234) eredménye 432! (a megoldás során ne használjon sztringeket/karaktertömböket)

31. Írjon függvényt vagy metódust, mely a paraméterként kapott 10x10-es mátrixról eldönti, hogy van-e olyan eleme, mely sorában nagyobb és oszlopában pedig kisebb a többi elemnél!
32. Írjon függvényt vagy metódust, mely visszaadja, hogy **k**-tól **m**-ig hány olyan szám van, melyeknek **n** db valódi osztója van! (**n**, **k** és **m** paraméter).
33. Írjon függvényt vagy metódust, mely visszaadja, hogy két pozitív egész paraméterének legkisebb közös többszöröse hány számjegyből áll kettes számrendszerben.