

*Filieră teoretică, profil real, specializare științe ale naturii*

- Toate subiectele sunt obligatorii. Se acordă zece puncte din oficiu.
- Timpul de lucru efectiv este de trei ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (bold), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.

**I TÊTEL**

(20 punct)

Az 1-től 5-ig számozott itemek esetén, írja a vizsgalapra a helyes válasznak megfelelő betűt. Minden helyes válasz 4 pontot ér.

- Adja meg azt az intervallumot, amelybe az **x** változóba tárolt érték tartozik, akkor és csakis akkor, ha a mellékelt C/C++ kifejezés értéke 1.  
`!(x<=2020) && !(x>2025)`  
a. (2020, 2025]      b. (2020, 2025)      c. [2020, 2025]      d. (2021, 2025)
- Egy egydimenziós tömb elemei, ebben a sorrendben (2, 3, 5, 7, 10, 12, 16, 17, 20, 21, 27). Hogy eldönthessük, hogy a tömbben megtalálható-e az **x** érték, bináris keresés módszerét használjuk. Adjon meg három olyan lehetséges értékét az **x**-nek úgy, hogy a keresés befejeződhessen miután az **x** össze lett hasonlítva a tömb három vagy négy elemével.  
a. 2, 3, 5      b. 2, 7, 21      c. 7, 16, 20      d. 20, 21, 27
- Adja meg a mellékelt C/C++ kifejezés értékét.  
`abs(-25)-25`  
a. -50      b. -25      c. 0      d. 50
- A mellékelt C/C++ szekvenciában az összes változó értéke egész, és az **n** ≥ 2. Adja meg a pontozott részt helyettesítő kifejezést, úgy hogy a kapott szekvencia végrehajtása után, a **x** változó értéke egyenlő legyen az **n** változóban tárolt érték legnagyobb osztójával, amely szigorúan kisebb mint **n**.  
`x=2; while(n%x!=0) x=x+1; x=.....;`  
a. **x-1**      b. **x\*x**      c. **n/2**      d. **n/x**
- Az **s1**, **s2** és **s3**-al jelzett szekvenciák meghatározása mellékelve van, ahol az összes változó egész. Adja meg azokat a szekvenciákat, amelyek különálló végrehajtása során, az **s** változó tárolja az 1·2+2·3+3·4+4·5+5·6+6·7+7·8+8·9+9·10 összeget.  

<code>//s1 s=0; for(i=10;i&gt;1;i--)   s=s+i*(i-1);</code>	<code>//s2 s=0; i=1; do { p=i*(i+1); s=s+p; i=i+1;   }while(i&lt;10);</code>	<code>//s3 s=0; i=1; p=1; while(i&lt;10) { p=p*(i+1); s=s+p; i=i+1;}</code>
--	--	---

  
a. **s1, s2**, de nem az **s3**      b. **s1, s3**, de nem az **s2**      c. **s2, s3**, de nem az **s1**      d. **s1, s2 és s3**

**II TÊTEL**

(40 pont)

- Adott a mellékelt algoritmus pszeudokódban.  
Az **a%b** az **a** természetes számnak a **b** nullától különböző természetes számmal való osztási maradékát és **[c]** a **c** valós szám egész részét jelöli.  
a. Írja le, mi kerül kiírásra az algoritmus végrehajtása nyomán, ha a beolvasott számok, ebben a sorrendben 9767 és 8204. (6p.)  
b. Ha az első beolvasott szám 2025, adjon meg két különböző értéket a **[10, 99]** intervallumból, amelyet az **y**-ba olvashatunk úgy, hogy mindegyik esetén, az algoritmus által kiírt üzenet a **DA** legyen, egy szám értékkel kiegészítve. (6p.)  
c. Írja meg az adott algoritmusnak megfelelő C/C++ programot. (10p.)  
d. Írjon pszeudokódban az adott algoritmussal egyenértékű algoritmust, helyettesítve az első **amíg...végezd el** struktúrát, hátul tesztelő ismétlődő struktúrával. (6p.)

```
beolvas x,y
(különböző természetes számok)
cx←0; cy←0
amíg x≥10 vagy y≥10 végezd el
  amíg x+y≠0 végezd el
    cx←cx+x%10; x←[x/10]
    cy←cy+y%10; y←[y/10]
  x←cx; cx←0; y←cy; cy←0
ha x=y akkor kiír "DA ",x
különbön kiír "NU ",x," ",y
```

2. Az  $A = (2, 5, 7, 9)$  és  $B = (x, y, z)$  egydimenziós táblák növekvő sorrendbe lettek összefésülve, balról jobbra való bejárással. Írjon egy, egymástól különböző természetes számokból álló példát az  $x$ ,  $y$  és  $z$ -re, tudva azt, hogy a két tömbnek nincsenek közös elemei, és hogy az  $x$  az  $A$  egyetlen elemével hasonlítódik össze, a  $z$  pedig nem hasonlítódik össze az  $A$  egyetlen elemével sem. (6p.)
3. Egy ékszer egyetlen nemesfémről készül, és az  $mb$  és  $gb$  egész típusú változók annak a fémnek a kódját tárolják, amelyből az ékszer készült, valamint annak súlyát, grammban kifejezve. Ismertek két nemesfém adatai, az  $m1$  és  $p1$  egész típusú változók tárolják az első nemesfém kódját, valamint egy gramm árát ebből a fémről, és az  $m2$  és  $p2$  egész típusú változók tárolják a második nemesfém kódját, valamint egy gramm árát a fémről. Írjon egy C/C++ szekvenciát, amelynek végrehajtása során a képernyőre kiírható az adott ékszer ára, ha az ékszer a két ismert nemesfém valamelyikből készült, vagy a `pret necunoscut` üzenet, ellenkező esetben. (6p.)

### III TÉTEL

(30 pont)

1. Egy farmernek téglalap alakú földterülete van, amelyet egyenlő területű parcellákra szeretne osztani, úgy hogy a parcellák száma és területe természetes számok legyenek, valamint a parcellák száma páros és szigorúan kisebb legyen, mint a parcellák területe. Beolvassunk két természetes számot a  $[2, 100]$  intervallumból,  $x$  és  $y$ , amelyek a terület hosszúságát és szélességét jelölik, méterben kifejezve. Írja ki az összes lehetséges megoldást a földterület felosztására, minden megoldást egy-egy sorba, két szám formájában, a `parcele de arie` szöveggel és a megfelelő szóközzel elválasztva, ahol az első szám a kapott parcellák számát jelöli, a második pedig egy megfelelő parcella területét, ahogy a példa mutatja. Ha nincs egyetlen ilyen megoldás sem, akkor a `nu exista` üzenetet írjuk ki. Írja le pszeudokódban a probléma megoldási algoritmusát. Az új sorra lépés a következőképpen jelölhető az algoritmusban: **kiír új sor**.

**Példa:** ha  $x=6$  és  $y=8$ , a mellékelt értékek lesznek kiírva, nem kötelezően ebben a sorrendben (a földet feloszthatjuk 3 darab 16 területű parcellára és 8 darab 6 területű parcellára, de ezek nem felelnek meg az elvárásoknak)

2	parcele de arie	24
4	parcele de arie	12
6	parcele de arie	8

(10p.)

2. Írjon C/C++ programot, amely beolvas a billentyűzetről egy természetes számot,  $n$  ( $n \in [2, 10^2]$ ), és egy  $n$  különböző természetes számból álló sorozatot a  $[0, 10^9]$  intervallumból, az egydimenziós tömb elemeit. A program módosítja a tömböt a memóriában, kicserélve a legkisebb értéket, az utolsó pozíción található értékre. A kapott tömb elemeit kiírjuk a képernyőre, szóközzel választva el egymástól.

**Például:** ha  $n=8$  és a tömb (17, 19, 10, 4, 23, 3, 11, 9)  
a kapott tömb (17, 19, 10, 4, 23, 9, 11, 9)

(10p.)

3. Egy napelempark esetén figyelik a **napi** és **össz**-energiatermelést, minden időszakra. A megfigyelt napok egymásutáni természetes számokkal vannak jelölve, kronológiai sorrendben, az 1-es naptól kezdődően. Egy időszak legalább két egymásutáni megfigyelési napból áll, és a rá vonatkozó össz-energiatermelést a megfelelő napi energiatermelések összegeként számítjuk ki. Egy **nap validált**, ha a napi termelés legalább egyenlő a napi határértékkel, a **minZ**-vel; egy **időszak validált**, ha a rá vonatkozó össztermelés legalább egyenlő a rövidtávú határértékkel, a **minP**-vel, minden napja validált és az időszak maximális hosszúságú ezzel a tulajdonsággal (nem adhatunk validált napot hozzá.)

A **bac.in** állomány legtöbb  $10^6$  természetes számot tartalmaz az  $[1, 10^3]$  intervallumból: az első sorban a **minZ** és a **minP**, a leírt határértékeknek megfelelően, a második sorban a megfigyelés sorrendjében, az egymásutáni napok napi energiatermelései. Az azonos sorban levő számokat szóközzel választjuk el.

Írja ki a képernyőre, minden, az adatoknak megfelelő validált termelési időszakra, az össztermelési mennyiséget. Az adatokat a képernyő egy-egy sorára írjuk ki, ha nem létezik egyetlen ilyen időszak sem, a `nu exista` üzenetet. Tervezen a futási idő és a felhasznált memória szempontjából hatékony algoritmust.

**Példa:** ha az állomány a mellékelt adatokat tárolja, a képernyőre kiírt értékek

65  
71  
90

10	40
65	9 20 25 12 14 7 3 11 15 12 8 19 50 21

a. Írja le saját szavaival a használt algoritmust és indokolja annak hatékonyságát. (2p.)

b. Írja meg a leírt algoritmusnak megfelelő C/C++ programot. (8p.)