

BBTE Matek-Infó verseny – 1. tételsor
INFORMATIKA írásbeli

A versenyzők figyelmébe:

1. A tömböket 1-től kezdődően indexeljük.
2. A rácsesztekre (A rész) egy vagy több helyes válasz lehetséges. A válaszokat a vizsgadolgozatba írjátok (nem a feladatlpra). Ahhoz, hogy a feltüntetett pontszámot megkapjátok, elengedhetetlenül szükséges, hogy minden helyes választ megadjatok, és kizárólag csak ezeket.
3. A B részben szereplő feladatok megoldásait részletesen kidolgozva a vizsgadolgozatba írjátok.
 - a. A feladatok megoldásait *pseudokódban* vagy egy *programozási nyelvben* (Pascal/C/C++) kell megadni.
 - b. A megoldások értékelésekor az első szempont az algoritmus *helyessége*, majd a *hatékonysága*, ami a *végrehajtási időt* és a *felhasznált memória méretét* illeti.
 - c. A tulajdonképpeni megoldások előtt, *kötelezően leírátok szavakkal az alprogramokat, és megindokoljátok a megoldások lépéseit*. Feltétlenül írjátok *megjegyzéseket* (kommenteket), amelyek segítik az adott megoldás technikai részleteinek megértését. Adjátok meg az azonosítók jelentését és a fölhasznált adatszerkezeteket. Ha ez hiányzik, a tételre kapható pontszámotok 10%-kal csökken.
 - d. Ne használjátok különleges fejláncokat, előredefiniált függvényeket (például STL, karakterláncokat feldolgozó sajátos függvények stb.).

A rész (30 pont)

A.1. Vajon mit csinál? (5p)

Adott az $\text{alg}(x, b)$ algoritmus (alprogram), amelynek bemeneti paraméterei az x és b természetes számok ($1 \leq x \leq 1000$, $1 < b \leq 10$). Határozzátok meg, hogy mit csinál az algoritmus.

```
Algoritmus alg(x, b):  
  s ← 0  
  Amíg x > 0 végezd el  
    s ← s + x MOD b  
    x ← x DIV b  
  vége(amíg)  
  térítsd s MOD (b - 1) = 0  
Vége(algoritmus)
```

- a. kiszámítja az x természetes szám számjegyeinek összegét a b számrendszerben
- b. vizsgálja, hogy az x szám számjegyeinek összege a $b - 1$ számrendszerben osztható-e $(b - 1)$ -gyel
- c. vizsgálja, hogy az x szám osztható-e $(b - 1)$ -gyel
- d. vizsgálja, hogy a b számrendszerben felírt x szám számjegyeinek összege osztható-e $(b - 1)$ -gyel
- e. vizsgálja, hogy az x szám számjegyeinek összege osztható-e $(b - 1)$ -gyel

A.2. Mit fog kiírni? (5p)

Legyen a következő program:

Varianta C++/C

```
int sum(int n, int a[], int s){  
  s = 0;  
  int i = 1;  
  while(i <= n){  
    if(a[i] != 0) s += a[i];  
    ++i;  
  }  
  return s;  
}  
  
int main(){  
  int n = 3, p = 0, a[10];  
  a[1] = -1; a[2] = 0; a[3] = 3;  
  int s = sum(n, a, p);  
  cout << s << ";"; << p; // printf("%d;%d", s, p);  
  return 0;  
}
```

Varianta Pascal

```
type vector = array[1..10] of integer;  
  
function sum(n:integer; a:vector; s:integer):integer;  
  var i : integer;  
  begin  
    s := 0; i := 1;  
    while (i <= n) do  
      begin  
        if (a[i] <> 0) then s := s + a[i];  
        i := i + 1;  
      end;  
    sum := s;  
  end;  
var n, p, s : integer;  
    a : vector;  
begin  
  n := 3; a[1] := -1; a[2] := 0; a[3] := 3; p := 0;  
  s := sum(n, a, p);  
  writeln(s, ';', p);  
end.
```

Mit fog kiírni a program?

- a. 0;0
- b. 2;0
- c. 2;2
- d. Egyik válasz sem helyes
- e. 0;2

A.3. Logikai kifejezés (5p)

Legyen a következő logikai kifejezés: $(X \text{ OR } Z) \text{ AND } (\text{NOT } X \text{ OR } Y)$. Válasszátok ki X, Y, Z értékeit úgy, hogy a kifejezés értéke legyen TRUE:

- $X \leftarrow \text{FALSE}; Y \leftarrow \text{FALSE}; Z \leftarrow \text{TRUE};$
- $X \leftarrow \text{TRUE}; Y \leftarrow \text{FALSE}; Z \leftarrow \text{FALSE};$
- $X \leftarrow \text{FALSE}; Y \leftarrow \text{TRUE}; Z \leftarrow \text{FALSE};$
- $X \leftarrow \text{TRUE}; Y \leftarrow \text{TRUE}; Z \leftarrow \text{TRUE};$
- $X \leftarrow \text{FALSE}; Y \leftarrow \text{FALSE}; Z \leftarrow \text{FALSE};$

A.4. Számolás (5p)

Legyen a számol(a, b) algoritmus, amelynek bemeneti paraméterei az a és b pozitív természetes számok, ahol $1 \leq a \leq 100, 1 \leq b \leq 100$.

1.	Algoritmus számol(a, b):
2.	Ha $a \neq 0$ akkor
3.	térítsd számol($a \text{ DIV } 2, b + b$) + $b * (a \text{ MOD } 2)$
4.	vége(ha)
5.	térítsd 0
6.	Vége(algoritmus)

Az alábbi válaszok közül melyek hamisak?

- ha a és b egyenlők, az algoritmus a értékét téríti
- ha $a = 1000$ és $b = 2$, az algoritmus 10-szer hívja meg önmagát
- az algoritmus által kiszámított és térített érték egyenlő $(a / 2 + 2 * b)$ -vel
- az 5. sorban található utasítás egyszer sem hajtódik végre
- az 5. sorban található utasítás egyszer hajtódik végre

A.5. Elem beazonosítása(5p)

Legyen az (1, 2, 3, 2, 5, 2, 3, 7, 2, 4, 3, 2, 5, 11, ...) sorozat, amelyet a következőképpen hoztunk létre: kiindulva természetes számok sorozatából, azokat a számokat, amelyek nem prímszámok helyettesítettük a saját osztóikkal úgy, hogy minden d osztót csak egyszer használtunk minden szám esetében. Az alábbi algoritmusok közül melyik határozza meg a sorozat n -dik elemét (n természetes szám, $1 \leq n \leq 1000$)?

a.	Algoritmus beazonosítás(n): $a \leftarrow 1, b \leftarrow 1, c \leftarrow 1$ Amíg $c < n$ végezd el $a \leftarrow a + 1, b \leftarrow a, c \leftarrow c + 1, d \leftarrow 2$ Amíg $c \leq n$ és $d \leq a \text{ DIV } 2$ végezd el Ha $a \text{ MOD } d = 0$ akkor $c \leftarrow c + 1, b \leftarrow d$ vége(ha) $d \leftarrow d + 1$ vége(amíg) vége(míg) térítsd b Vége(algoritmus)	b.	Algoritmus beazonosítás(n): $a \leftarrow 1, b \leftarrow 1, c \leftarrow 1$ Amíg $c < n$ végezd el $c \leftarrow c + 1, d \leftarrow 2$ Amíg $c \leq n$ és $d \leq a \text{ DIV } 2$ végezd el Ha $a \text{ MOD } d = 0$ akkor $c \leftarrow c + 1, b \leftarrow d$ vége(ha) $d \leftarrow d + 1$ vége(amíg) $a \leftarrow a + 1, b \leftarrow a$ vége(amíg) térítsd b Vége(algoritmus)
c.	Algoritmus beazonosítás(n): $a \leftarrow 1, b \leftarrow 1, c \leftarrow 1$ Amíg $c < n$ végezd el $a \leftarrow a + 1, d \leftarrow 2$ Amíg $c < n$ és $d \leq a$ végezd el Ha $a \text{ MOD } d = 0$ akkor $c \leftarrow c + 1, b \leftarrow d$ vége(ha) $d \leftarrow d + 1$ vége(amíg) vége(amíg) térítsd b Vége(algoritmus)	d.	Algoritmus beazonosítás(n): $a \leftarrow 1, b \leftarrow 1, c \leftarrow 1$ Amíg $c < n$ végezd el $b \leftarrow a, a \leftarrow a + 1, c \leftarrow c + 1, d \leftarrow 2$ Amíg $c \leq n$ és $d \leq a \text{ DIV } 2$ végezd el Ha $a \text{ MOD } d = 0$ akkor $c \leftarrow c + 1, b \leftarrow d$ vége(ha) $d \leftarrow d + 1$ vége(amíg) vége(amíg) térítsd b Vége(algoritmus)
e.	Algoritmus beazonosítás(n): $a \leftarrow 1, b \leftarrow 1, c \leftarrow 1$ Amíg $c < n$ végezd el $a \leftarrow a + 1, b \leftarrow a, c \leftarrow c + 1, d \leftarrow 2$ $f \leftarrow \text{false}$ Amíg $c \leq n$ és $d \leq a \text{ DIV } 2$ végezd el Ha $a \text{ MOD } d = 0$ akkor $c \leftarrow c + 1, b \leftarrow d, f \leftarrow \text{true}$ vége(ha) $d \leftarrow d + 1$ vége(amíg) Ha f akkor $c \leftarrow c - 1$ vége(ha) vége(amíg) térítsd b Vége(algoritmus)		

A.6. Törzstényezők (5p)

Legyen a törzstényezők(n , d , k , x) algoritmus, amely meghatározza az n természetes szám k darab törzstényezőjét, a törzstényezők keresését a d értéktől kezdve. Bemeneti paraméterek az n , d és k számok, kimeneti paraméterek az x sorozat, amely a k törzstényezőt tartalmazza ($1 \leq n \leq 10000$, $2 \leq d \leq 10000$, $0 \leq k \leq 10000$).

```
Algoritmus törzstényezők( $n$ ,  $d$ ,  $k$ ,  $x$ ):  
  Ha  $n \text{ MOD } d = 0$  akkor  
     $k \leftarrow k + 1$   
     $x[k] \leftarrow d$   
  vége(ha)  
  Amíg  $n \text{ MOD } d \neq 0$  végezd el  
     $n \leftarrow n \text{ DIV } d$   
  vége(amíg)  
  Ha  $n > 1$  akkor  
    törzstényezők( $n$ ,  $d + 1$ ,  $k$ ,  $x$ )  
  vége(ha)  
Vége(algoritmus)
```

Határozzátok meg, hányszor hívja meg önmagát a törzstényezők(n , d , k , x) algoritmus a következő programrészlet végrehajtásának következtében:

```
 $n \leftarrow 120$   
 $d \leftarrow 2$   
 $k \leftarrow 0$   
törzstényezők( $n$ ,  $d$ ,  $k$ ,  $x$ )
```

- 3-szor
- 5-ször
- 9-szer
- 6-szor
- ugyanannyiszor, mint a következő programrészlet esetében

```
 $n \leftarrow 750$   
 $d \leftarrow 2$   
 $k \leftarrow 0$   
törzstényezők( $n$ ,  $d$ ,  $k$ ,  $x$ )
```

B rész (60 pont)

B.1. Konverzió (10 pont)

Legyen a konverzió(s , $hossz$) algoritmus, amely átalakítja az s karakterláncot, amely egy 16-os számrendszerben ábrázolt szám, a megfelelő 10-es számrendszerben érvényes alakjára. Az s karakterlánc $hossz$ darab karaktert tartalmaz, ahol a karakter értéke lehet egy '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' számjegy, vagy egy 'A', 'B', 'C', 'D', 'E', 'F' nagybetű (a $hossz$ természetes szám, $1 \leq hossz \leq 10$).

Írjátok le a konverzió(s , $hossz$) algoritmus *rekurzív* változatát úgy, hogy a fejléce és a hatása legyen azonos az alábbi algoritmusával:

```
Algoritmus konverzió( $s$ ,  $hossz$ ):  
  szám  $\leftarrow 0$   
  Minden  $i \leftarrow 1$ ,  $hossz$  végezd el  
    Ha  $s[i] \geq 'A'$  akkor  
      szám  $\leftarrow szám * 16 + s[i] - 'A' + 10$   
    különben  
      szám  $\leftarrow szám * 16 + s[i] - '0'$   
  vége(ha)  
  vége(minden)  
  térítsd szám  
Vége(algoritmus)
```

B.2. Azonos számjegyek (20 pont)

Adott két természetes szám: a és b , ahol $1 \leq a \leq 1\,000\,000$ és $1 \leq b \leq 1\,000\,000$.

Írjatok algoritmust, amely meghatározza a k elemű x sorozatot, (k – természetes szám, $0 \leq k \leq 1000$), amely minden olyan természetes számot tárol, amelyek az $[a, b]$ intervallumhoz tartoznak és azonos számjegyekből állnak. Ha ilyen szám nem létezik, k értéke 0 lesz. Az algoritmus bemeneti paraméterei a és b , kimeneti paraméterek pedig k és x .

1. Példa: ha $a = 8$ és $b = 120$, akkor $k = 12$ és $x = (8, 9, 11, 22, 33, 44, 55, 66, 77, 88, 99, 111)$.

2. Példa: ha $a = 590$ és $b = 623$, akkor $k = 0$ és az x sorozat üres.

B.3. Sétáló robot (30 pont)

Egy robot egy négyzetes tömb alakú térképen mozog, amelynek a mérete páratlan szám, és a térkép minden celláján letesz bizonyos számú tárgyat. A robot a következő szabályok alapján mozog:

- a cellában, ahonnan indul letesz egy tárgyat, a második cellában, ahova lép, letesz kettőt, a harmadik cellában, ahova eljut, hármat és így tovább;
- a robot az utolsó oszlop közepéről indul, és lép egyet átlósan a felfele és jobbra szomszédos szabad cellára (párhuzamosan a mellékatlóval) ha ez a cella létezik és szabad a hely; ha ilyen cella nincs, akkor:
 - ha a robot az utolsó oszlopban található, akkor „átugrik” az első oszlopba, a robot felett levő sorba, ha ez a hely szabad;
 - ha a robot az első sorban található, akkor „átugrik” az utolsó sorba a robottól jobbra levő oszlopba, ha ez a hely szabad;
 - ha a robot a térkép jobb-felső sarkában található. akkor átugrik az utolsó sor első oszlopában levő cellába, ha ez a hely szabad.
- ha a cella, ahova lépni szeretne foglalt, a robot balra lép egyet a sorban, ahol éppen található, a szomszédos szabad cellába.

Ezek a szabályok biztosítják, hogy a robot a térkép minden celláját egyszer látogatja meg (és azt is, hogy nem fog elakadni sehol). Miután a robot a térkép minden cellájában elhelyezte a tárgyakat, megáll.

Például, egy 5×5 cellából álló térképen a robot első 22 lépése a következő:

9	3	22	16	15
2	21	20	14	8
	19	13	7	1
18	12	6	5	
11	10	4		17

Írjatok algoritmust, amely meghatározza azoknak a tárgyaknak a *szám* számát, amelyeket a robot a térkép főátlóján levő cellákba helyez. Az algoritmus bemeneti paramétere a térkép n mérete (n – páratlan természetes szám, $3 \leq n \leq 100$), kimeneti paramétere a *szám* (*szám* – természetes szám).

1. *Példa*: ha $n = 5$, akkor *szám* = 65.

2. *Példa*: ha $n = 11$, akkor *szám* = 671.

Megjegyzések:

1. Minden tétel kidolgozása kötelező.
2. A piszkozatokat nem vesszük figyelembe.
3. Hivatalból jár 10 pont.
4. Rendelkezésekre áll 3 óra.

JAVÍTÓKULCS

HIVATALBÓL	10 pont
A rész.....	30 pont
A. 1. c, d	5 pont
A. 2. b	5 pont
A. 3. a, d	5 pont
A. 4. a, c, d	5 pont
A. 5. e	5 pont
A. 6. a, e	5 pont
B rész.....	60 pont
B. 1. Konverzió	10 pont
– fejléc a követelményeknek megfelelően	2 pont
– a rekurzív hívások leállási feltétele.....	1 pont
– a leálláskor térített érték.....	1 pont
– nem számjegy karakter feltétele	2 pont
– térített érték, amikor a karakter nem számjegy	2 pont
– térített érték, amikor a karakter számjegy	2 pont

```
Algoritmus konverzió(s, hossz):
  Ha hossz > 0 akkor
    Ha s[hossz] ≥ 'A' akkor
      térítsd konverzió(s, hossz - 1) * 16 + s[hossz] - 'A' + 10
    különben
      térítsd konverzió(s, hossz - 1) * 16 + s[hossz] - '0'
  vége(ha)
különben
  térítsd 0
vége(ha)
Vége(algoritmus)
```

B. 2. Azonos számjegyek.....	20 pont
• V1: az azonos számjegyeket tartalmazó számok – számolásokkal történő generálása	20 pont
○ bemeneti és kimeneti paraméterek a követelményeknek megfelelően	2 pont
○ a számok generálása (egyszámjegyű számok és/vagy 11, 111, 1111, többszörösei...)	16 pont
○ az azonos számjegyű számok sorozatba mentése	2 pont
• V2: az $[a, b]$ intervallum számainak bejárása és vizsgálata	10 pont
○ bemeneti és kimeneti paraméterek a követelményeknek megfelelően	2 pont
○ annak a tulajdonságnak az ellenőrzése, hogy a szám számjegyei azonosak.....	4 pont
○ az $[a, b]$ intervallumhoz tartozó számok bejárása az ellenőrzés céljából	2 pont
○ az azonos számjegyű számok sorozatba mentése	2 pont
B. 3. Sétáló robot.....	30 pont
• V1: a kért eredmény helyes meghatározása számolással: $(n*n*n+n)/2$	30 pont
○ bemeneti és kimeneti paraméterek a követelményeknek megfelelően	2 pont
○ számolás.....	14 pont
○ a számolások részletes megindoklása	14 pont
• V2: a kért eredmény helyes meghatározása szimulálással	25 pont
○ bemeneti és kimeneti paraméterek a követelményeknek megfelelően	2 pont
○ az összes $n \times n$ cella bejárása	4 pont
○ helyes mozgás (a 4 lehetséges esetben)	4*4 pont
○ a főátlón található tárgyak számának helyes meghatározása	3 pont

B rész (megoldások) 60 pont

B. 1. Konverzió 10 pont

```
Algoritmus konverzió(s, hossz):
  Ha hossz > 0 akkor
    Ha s[hossz] ≥ 'A' akkor
      térítsd konverzió(s, hossz - 1) * 16 + s[hossz] - 'A' + 10
    különben
      térítsd konverzió(s, hossz - 1) * 16 + s[hossz] - '0'
    vége(ha)
  különben
    térítsd 0
  vége(ha)
Vége(algoritmus)
```

B. 2. Azonos számjegyek..... 20 pont

- V1: az azonos számjegyeket tartalmazó számok – számolásokkal történő generálása 20 pont

```
void azonosSzamjegyek(int a, int b, int &k, int x[]){
  k = 0;
  for(int i = a; ((i < 10) && (i <= b)); i++){
    x[++k] = i;
  }
  int akt = 11;
  int leptek = 11;
  while(akt <= b){
    if(akt >= a)
      x[++k] = akt;
    akt = akt + leptek;
    if(akt > 9 * leptek){
      akt = leptek * 10 + 1;
      leptek = akt;
    }
  }
}
```

B. 3. Sétáló robot..... 30 pont

- V1: a kért eredmény helyes meghatározása számolással: $(n*n*n+n)/2$ 30 pont

9	3			
2				8
		13	7	1
	12	6	5	
11	10	4		

17	23			
24				18
		13	19	25
	14	20	21	
15	16	22		

		22	16	15
	21	20	14	
25	19	13		
18				24
			23	17

9	3	22	16	15
2	21	20	14	8
25	19	13	7	1
18	12	6	5	
11	10	4		17

A robot útvonala szimmetrikus a négyzet közepéhez viszonyítva. Másképp fogalmazva, a négyzet közepét a robot pontosan az útnak a közepén érinti $((n*n - 1)/2)$ lépés után). Az útvonal második része ugyanaz, mint az első, de a bejárást fordított sorrendben végezzük és elforgatjuk 180 fokkal a négyzet közepéhez viszonyítva (például, az utolsó négyzet a bal szél közepén lesz). Következik, hogy bármely négyzetszámra, amelyek szimmetrikusak a középponthoz viszonyítva, fennáll, hogy a tárgyak számának összege $n*n + 1$. Ha ezt az észrevételt alkalmazzuk a főátló elemeire, majd párokat alkotunk belőlük, megkapjuk a kért összeget az $n*(n*n + 1)/2$ kifejezést.

Ha $n = 5$, a térítendő összeg 65.

```
int targyak(int n){
  return (n * n * n + n) / 2;
}
```