Pótórák feladatai és megoldásai

1. Pótóra (2024. Márcuis 05.)

A feladat (hátultesztelő)

1. Kérj a felhasználótól egy 3 jegyű egész számot.
2. Ha nem ilyet ír be, kérd újra.
3. Ha megfelelő számot írt be, alkoss az utolsó számjegyéből egy kétjegyű számot, ami a 11 többszöröse, és az annál 7-tel nagyobbat.
   `PL 635 -> 55 és 62`
   Megoldás

```cpp
#include <iostream>

using namespace std;

int main() {
    int sz;

    // azt kell átgondolni, hogy egyszer megengedhetjük-e,
    // hogy lefusson a do ciklus
    do {
        cout << "Kérek egy háromjegyű számot! " << endl;
        cin >> sz;
    } while (sz < 100 || sz > 999);

    // ugyenezt elérhetnénk a következő módon is:
    /*
    cout << "Kérek egy háromjegyű számot! " << endl;
    cin >> number;
    while (number < 100 || number > 999) {
        cout << "Kérek egy háromjegyű számot! " << endl;
        cin >> number;
    }
    */
   // ugyanazt értük el, viszont több sorban kellett leírnunk.
   // mivel egyszer mindenképp lefut az adatbekérés (16-17. sor) (bármilyen

    // A feladat hátralevő része már cikluson kívül oldható meg,
    // mivel nem kell minden megadott adatra elvégezni a műveleteket.
    // Ha ez is a ciklusokon belül lenne, akkor feleslegesen végeznénk el e
    // Például:
    // Ha szeretnél egy almát felvágni négybe, és sorba adogatnak a kezedbe
    // Inkább kérsz egy másik gyümölcsöt, mindaddig, amíg az nem alma. Ekko
    int tizenegyszer = number % 10 * 10 + number % 10;
    int pluszHet = tizenegyszer + 7;

    cout << "Az utolsó számjegyből alkotott 11 többszöröse " << tizenegysze
    return 0;
}
```

B feladat (hátultesztelő)

1. Kérj a felhasználótól egy kisbetűt!
2. Ha nem kisbetűt ír be, akkor kérd újra!
3. Ha kisbetűt írt be, akkor írjuk ki a nagybetűs párját, háromszor egymás alá, úgy hogy legyen közöttük üres sor!
   Megoldás

```cpp
#include <iostream>

using namespace std;

int main() {
    char betu;
    do {
        cout << "Kérek egy kisbetűt! " << endl;
        cin >> betu;
    } while (betu < 97 || betu > 122);
    // ascii táblázatból láthatjuk, hogy a kisbetűk kódjai 97 és 122 között
    // Az első feladathoz hasonlóan a do ciklus használata itt indokolt,
    // mivel egyszer mindenképp lefut az adatbekérés.

    // Ahhoz, hogy a kis a-ból nagy A legyen, meg kell keresnünk,
    // hogy mekkora a kettő közt a távolság.
    // A kis a kódja 97, a nagy A kódja 65, tehát 32 a különbség.
    // Mivel párhuzmosan növekednek a számok,
    // ezért ez a különbség nem csak az a-ra igaz,
    // hanem minden betűre.

    char nagyBetu = betu - 32;

    //mivel háromszor kell kiírni, kell egy for ciklus, ami háromszor fut l
    for (int i = 0; i < 3; i++) {
        cout << nagyBetu << endl;
        // A feladat kér közéjük egy-egy üres sort is,
        // ezért nem elég egy endl, kettő kell.
        cout << endl;
    }

    return 0;
}
```

C feladat (elöltesztelő)

1. Kérj a felhasználótól egy egyjegyű számot, és őrizd meg az `n` változóban!
2. Amíg a szám kisebb mint `100`, adj hozzá véletlen számokat a `[10,50]`

intervallumból.

3. Ha az `n` eléri a 100-at, lépj ki a ciklusból!
4. Menet közben írasd ki a generált számokat (nem az összeget) egymás mellé, szóközzel elválasztva
5. Menet közben számold meg, hány számot generáltál ki
6. Menet közben számítsd ki a generált számok összegét
7. Menet közben számold meg, hány `1`-gyel kezdődő számot generáltál
8. Írj üzenetet az eredményekről
9. Számítsd ki és írd ki a generált számok átlagát!

   Megoldás

```cpp
#include <iostream>

using namespace std;

int main() {
    srand(time(0));

    cout << "Kérek egy egyjegyű számot! " << endl;
    int n;
    cin >> n;
    int szamlal = 0;
    int sum = 0;
    int szamlalEgyes = 0;

    // Mivel a feladat nem kéri, hogy ellenőrizzük a megadott számot,
    // ezért nem garantált, hogy az tényleg egyjegyű lesz.
    // Ezért ha a felhasználó nem fogad szót,
    // és rögtön 100-nál nagyobb számot ad meg,
    // akkor egyszer sem adhatunk hozzá véletlen számot.
    // Emiatt, ha do ciklust használnánk, akkor nem teljesítenénk a feladat
    /*
    do {
        int random = rand() % 41 + 10;
        szamlal++;
        sum += random;
        if (random / 10 == 1) {
            szamlalEgyes++;
        }
        cout << random << " ";
        n += random;
    } while (n < 100);
    */
    // Így még ha 100-nál nagyobb számot kapunk, akkor is adtunk hozzá véle
    // (pedig nem kellett volna), mivel a do ciklus legalább egyszer lefut.

    // Ezért inkább while ciklust használunk,
    // mivel a feltétel nem garantáltan teljesül a ciklus első futásakor.
    while (n < 100) {
        int random = rand() % 41 + 10;
```

```
            counter++;
            sum += random;
            if (random / 10 == 1) {
                szamlalEgyes++;
            }
            cout << random << " ";
            n += random;
        }

        cout << endl;
        cout << "Generált számok száma: " << szamlal << endl;
        cout << "Generált számok összege: " << sum << endl;
        cout << "1-gyel kezdődő számok száma: " << szamlalEgyes << endl;
        double avg = (double)sum / counter;
        cout << "Generált számok átlaga: " << avg << endl;

        return 0;
    }
```

D feladat

1. Írj függvényt, ami a kocka megadott élhosszából (cm) kiszámítja egy kocka térfogatát, (cm)!
   `int kocka (int a)`
2. A főprogramban kérd be a felhasználótól a kocka élhosszát cm-ben!
3. Hívd meg a térfogat számítására szolgáló függvényt!
4. Írasd ki, hogy hány köbcentiméter a kocka térfogata!
   Megoldás

```cpp
#include <iostream>

using namespace std;

int kocka (int a) {
    int terf = a * a * a;
    return terf;
}

int main() {
    cout << "Kérem a kocka élének hosszát! " << endl;
    int elhossz;
    cin >> elhossz;
    int terfogat = kocka(elhossz);
    cout << "A kocka térfogata " << terfogat << endl;
    return 0;
}
```

E feladat

1. Írj függvényt ami egy derékszögű háromszög két befogójából kiszámítja az átfogót!
   `double atfogo (double a, double b)`
2. A főprogramban kérd be a felhasználótól a befogók hosszát, tizedes számok is lehetnek!
3. Hívd meg a függvényt, és írd ki az eredményt!
   Megoldás

```cpp
#include <iostream>
#include <cmath>

using namespace std;

double atfogo (double a, double b) {
    double atfogo = sqrt(a * a + b * b);
    return atfogo;
}

int main() {
    cout << "Kérem a befogók hosszait! " << endl;
    double befogo1, befogo2;
    cin >> befogo1 >> befogo2;
    double atfogoHossz = atfogo(befogo1, befogo2);
    cout << "Az átfogó hossza " << atfogoHossz << endl;
    return 0;
}
```

2. Pótóra (2024. Március 12.)

A feladat

1. Az `a` tömb elemeinek add kezdőértékként: `{2, -10, 8, 3, 11, 20, 7}`
2. Írj függvényt, ami kiírja a tömb minden elemét egy sorba, szóközzel elválasztva! Az utolsó elem után kövekezzen 2 üres sor!
   `void kiirsorba(int *a, int n)`
   `a` - a tömb neve, `n` - az elemeinek a száma
3. A főprogramból hívd meg a függvényt és mutasd meg a tömb elemeit!
4. Deklarálj egy új tömböt, aminek a neve `haromszoros` legyen! (gondolkodj el, milyen típusú adatok kerülnek majd bele).
5. Az új tömbbe az a tömb elemeinek háromszorosa kerüljön!
6. Írasd ki a haromszoros tömb elemeit is a kész függvénnyel!
7. Írj függvényt, ami a összeadja egy tömb páros indexű elemeit!
   `int pindossz(int *a, int n)`
   `a` - a tömb neve, `n` - az elemeinek a száma

8. Hívd meg a függvényt a főprogramból és írasd ki mindkét tömb páros indexű elemeinek az összegét, üzenetek kíséretében!
9. Számítsd ki az eredeti tömb páros indexű elemeinek az átlagát is, és írasd ki!

Megoldás

```cpp
#include <iostream>

using namespace std;

void kiirsorba(int *a, int n) {
    for (int i = 0; i < n; i++) {
        cout << a[i] << " ";
    }
    cout << endl;  //ez az endl meg a kiiras veget jelzi
    cout << endl;
    coud << endl;
}

int pindossz(int *a, int n) {
    int sum = 0;
    for(int i = 0; i < n; i += 2) {
        sum += a[i];
    }
    return sum;
}

int main() {
    int a[7] = {2, -10, 8, 3, 11, 20, 7};

    kiirsorba(a, 7);

    int haromszoros[7];
    for (int i = 0; i < 7; i++) {
        haromszoros[i] = a[i] * 3;
    }

    kiirsorba(haromszoros, 7);

    cout << "Páros indexű elemek összege a tombben: " << pindossz(a, 7) << endl
    cout << "Páros indexű elemek összege a haromszoros tombben: " << pindossz(h

    cout << "Paros indexu elemek atlaga az a tombben: " << (float)pindossz(a, 7

    return 0;
```

```
}
```

B feladat

1. Deklarálj tömböt 10 egész szám tárolására!
2. Töltsd fel véletlen számokkal az `[10,205]` intervallumból! Addig végezd a feltöltést, míg a tömb tele nem lesz, vagy az első 100-asig. Tehát lehet ilyen tömböd: `{56, 200, 100}` vagy `{10,20,30,40, 60, 80, 90, 200, 199, 198}`, de nem lehet `{10,20,30,40, 60, 80, 90, 100, 199, 198}`. A ciklus típusát magad válaszd meg!
3. Számold meg, hogy hány elemet vittél be a tömbbe! Írasd is ki, miután kiléptél a ciklusból!
4. Írj függvényt, ami kiírja a képernyőre a tömb elemeit 3 oszlopba!
   `void kiir(int *a, int n)`
   `a` - a tömb neve, `n` - az elemeinek a száma
   Ügyelj arra, hogy a tomb végén lévő szemetet ne írasd ki, ha kevesebb mint 10 elem van!
5. Írj függvényt, ami megszámolja, hogy a tömbben hány kétjegyű szám van! A függvényt hívd meg helyesen a főprogramból, és írasd ki az eredményt!
   `int ketj(int *a, int n)`
   `a` - a tömb neve, `n` - az elemeinek a száma
   A főprogramból hívd meg a függvényt, írasd ki a kétjegyűek számát!
6. A főprogramban végezd a következő műveletet: a tömb minden páros elemét oszd el 2-vel! A páratlanokat ne változtasd!
7. Írasd ki a tömb elemeit, a már kész függvénnyel!
8. Írasd ki megint a kétjegyűek számát is! Használd a kész függvényt!
   Megoldás

```cpp
#include <iostream>

using namespace std;

void kiir(int *a, int n) {
    for (int i = 0; i < n; i++) {
        cout << a[i] << "\t";
        if(i % 3 == 2) {
            cout << endl;
        }
    }
    cout << endl;

}

int ketj(int *a, int n) {
    int db = 0;
    for (int i = 0; i < n; i++) {
        if( (a[i] > 10) && (a[i] < 100)) {
            db++;
        }
    }
    return db;
}

int main() {
    int a[10];

    srand(time(0));
    /*for (int i = 0; i < 10; i++) {
        int random = rand() % 196+10;
        if(random != 100) {
            a[i] = random; // 10-205
        }
        else {
            break;
        }

    }*/
```

```
    int random = rand() % 196+10;
    int ind = 0;
    while ((random != 100) && (ind < 10)) {
        int random = rand() % 196+10;
        a[ind] = random;
        ind++;
    }

    cout << ind << " elemet vittem be\n";

    kiir(a, ind);

    cout << "Ketjegyu szamok szama: " << ketj(a, ind) << endl;

    cout << "<---------------Osztas utan--------------->\n";

    for(int i = 0; i < ind; i++) {
        if (a[i] % 2 == 0) {
            a[i] = a[i] / 2;
        }
    }

    kiir(a, ind);

    cout << "Ketjegyu szamok szama: " << ketj(a, ind) << endl;

    return 0;
}
```

C feladat

1. Hozz létre új projektumot! A projektum mappájában hozz létre egy txt filet, aminek a neve `xx.txt` legyen! Írj bele egész számokat szóközzel elválasztva, legalább 8 db-ot! Őrizd meg a filet!
2. A programban deklarálj egy 15 elemű tömböt egész számok tárolására!
3. Nyisd meg a filet olvasásra, ellenőrizd, hogy helyesen nyílt e meg a file!
4. A file elemeit olvasd be a tömbbe! Ügyelj arra, hogy a tömb határát (15 elem) ne lépd túl!

5. Írasd ki, hogy hány elemet olvastál be a tömbbe!
6. Írj függvényt, ami kiszámítja az elemek összegét!
   `int osszeg(int *a, int n)`
   `a`-a tömb neve, `n` - az elemeinek a száma
7. A főprogramból hívd meg a függvényt, és írasd ki az összeget!
8. Számítsd ki az elemek átlagát is! Írasd ki!
9. Próbáld ki úgy is, hogy kevesebb mint 15 szám van a file-ban, és úgy is, hogy több!
   Megoldás

```cpp
#include <iostream>
#include <fstream>

using namespace std;

int ketj(int *a, int n) {
    int sum = 0;
    for(int i = 0; i < n; i++) {
        sum += a[i];
    }
    return sum;
}

int main() {
    int a[15];

    ifstream f("xx.txt");
    if(!f.is_open()) {
        cout << "Error opening file\n" << endl;
        return 1;
    }

    int i = 0;
    while(!f.eof() && (i < 15)) {
        f >> a[i];
        //cout << a[i] << " ";
        i++;
    }
    //cout << endl;

    cout << i << " darab szamot olvastam be\n";

    cout << "A szamok osszege: " << ketj(a, i) << endl;

    cout << "A szamok atlaga: " << (float)ketj(a, i) / i << endl;

    return 0;
}
```