

Makró

- 1.1. Készítsen egy makrót **ABS(x)** néven, amely a felhasználótól bekért érték abszolút értékét adja vissza. (2p)
- 1.2. Készítsen egy makrót **SQUARE(x)** néven, amely egy szám négyzetét adja vissza. (2p)
- 1.3. Készítsen egy makrót **IS_EVEN(y)** néven, amely a felhasználótól bekért tetszőleges értékről megállapítja, hogy páros-e vagy páratlan. (3p)
- 1.4. Készítsen egy makrót **IS_DIVISIBLE(x, y)** néven, amely meghatározza, hogy az első szám maradék nélkül osztható-e a másodikkal. (3p)
- 1.5. Készítsen egy makrót **CIRCLE_AREA(r)** néven, amely egy kör területét számítja ki. (3p)
- 1.6. Készítsen egy makrót **FACTORIAL(n)** néven, amely kiszámolja egy szám faktoriálisát. (4p)
- 1.7. Készítsen egy makrót **IS_TRIANGLE_POSSIBLE(a, b, c)** néven, amelyben három értéket kér be a felhasználótól, ezek lesznek az oldalhosszak. A makró logikai értéket adjon vissza annak függvényében, hogy a bekért három értékkel megrajzolható-e egy háromszög. (4p)
- 1.8. Készítsen egy makrót **IS_PRIME(n)** néven, amely meghatározza, hogy egy bekért szám prímszám-e. (5p)
- 1.9. Készítsen egy makrót **TO_UPPER_CASE(c)** néven, amely egy kisbetűs karaktert nagybetűvé alakít. (5p)

Függvény

- 2.1. Kérjen be a felhasználótól egy szög és egy átfogó értéket. Készítsen egy függvényt **haromszog** néven, amely a megadott értékek alapján derékszögű háromszöget feltételezve meghatározza a háromszög további oldalainak hosszát és a harmadik szögét, majd kiírja ezeket a felhasználónak. (4p)
- 2.2. Egy ékezetek nélküli karakterlánchoz készítsen egy függvényt **maganhangzok** néven, amely megszámolja és visszaadja a mondatban található magánhangzók számát. (4p)
- 2.3. Kérjen be a felhasználótól egy hőmérséklet értéket Celsiusban. Készítsen egy függvényt **celsius_fahrenheit** néven, amely átváltja a hőmérsékletet Fahrenheitre. (4p)
- 2.4. Kérjen be a felhasználótól egy évet. Készítsen egy függvényt **szokoev** néven, amely meghatározza, hogy az adott év szökőév-e, és visszaadja az eredményt. (4p)
- 2.5. Kérjen be a felhasználótól két számot. Készítsen egy függvényt **legnagyobb_kozos_oszto** néven, amely meghatározza és visszaadja a két szám legnagyobb közös osztóját. (5p)
- 2.6. Egy előre meghatározott mondathoz készítsen egy függvényt **nagybetu** néven, amely a mondat minden szavának kezdőbetűjét nagybetűvé alakítja (5p)
- 2.7. Kérjen be a felhasználótól egy számot. Készítsen egy függvényt **fibonacci** néven, amely visszaadja a megadott számhoz tartozó Fibonacci-számot. (5p)
- 2.8. Kérjen be a felhasználótól egy 8 bit-es logikai értéket (csak 0-t vagy 1-t tartalmazhat, pl.: 01101001). Készítsen egy függvényt **negalt** néven, amely ennek a logikai értéknek a negáltját adja vissza. A függvény írjon ki hibát, ha a felhasználó nem logikai értéket adott meg vagy a megadott érték nem 8 bit. (6p)
- 2.9. Kérjen be a felhasználótól egy listát (tömböt), amely számokat tartalmaz. Készítsen egy függvényt **rendezes** néven, amely rendezi a listát növekvő sorrendbe. (6p)

Pointer, láncolt lista

- 3.1. Készítsen egy függvényt **datum** néven, amely paraméterként (pointerként) megkapja egy számsorozat memóriacímét. A számsorozat egy dátum, formátuma legyen: ÉÉÉÉ/HH/NN. A függvény segítségével írja ki a felhasználónak az évet, a hónapot és a napot külön-külön. (5p)
- 3.2. Előre meghatározott 3 szót fűzzön egymás után, majd adja meg az összefűzött szöveg hosszát két módon (pointer és beépített függvény segítségével). (4p)
- 3.3. Készítsen egy függvényt **szorzat** néven, amely paraméterként (pointerként) megkapja egy egész számokat tartalmazó tömb memóriacímét és a tömb hosszát. A függvény számolja ki és adja vissza a tömb elemeinek szorzatát. (4p)
- 3.4. Készítsen egy függvényt **nevek** néven, amely paraméterként (pointerként) megkapja egy karakterlánc memóriacímét. A karakterlánc egy teljes nevet tartalmaz vezetéknév és keresztnév formátumban, pl.: "Kovács János". A függvény segítségével írja ki külön a vezetéknévet és a keresztnévet. (4p)
- 3.5. Készítsen egy függvényt **min_max** néven, amely paraméterként (pointerként) megkapja egy egész számokat tartalmazó tömb memóriacímét és a tömb hosszát. A függvény adja vissza a tömb legkisebb és legnagyobb elemét és ezek memóriacímét. (5p)
- 3.6. Készítsen egy függvényt **reverse_string** néven, amely egy karakterláncot kap paraméterként (pointerként), és visszaadja a megfordított változatát. (4p)
- 3.7. Készítsen egy 5 elemű láncolt listát, amely egy osztályban levő tanulók neveit és pontszámait tartalmazza. Készítsen egy függvényt, amely csökkenő sorrendbe rendezi a tanulókat a pontszámaik alapján. (5p)
- 3.8. Készítsen egy 5 elemű láncolt listát, amely egy osztályban levő tanulókat és magasságukat tartalmazza centiméterben. Készítsen egy függvényt, amely visszaadja az osztály átlagmagasságát és a legmagasabb tanulót. (5p)
- 3.9. Készítsen egy 5 elemű láncolt listát, amely egy osztályban levő tanulók keresztnéveit tartalmazza ékezetek nélkül. Rendezze ezeket ABC sorrendbe (csak a nevek kezdőbetűje számít). Készítsen egy függvényt, ami lehetővé teszi egy tetszőleges nevű, hatodik tanuló ABC sorrend szerint megfelelő helyre történő beszúrását. Listázza ki az eredeti és a hat elemű listát. (A nevek összehasonlításához használja a strcmp függvényt.) (7p)