

	Fájlnev/könyvtár	Feladat												
01.	8KIRALYNO.CPP	Készítsen programot, amely elhelyez 8 királynőt a sakktáblán! A program bemenete az első királynő sora. Jelenítse meg az elrendezést sematikusán!												
02.	AMSZOTAR.CPP	Fejlesszen angol-magyar szótárprogramot, ahol a szavakhoz a hashkódjukat is tárolja! Időméréssel hasonlítsa össze a fordítás sebességét szöveges, illetve hashtábla alapján történő keresés mellett! Ugyancsak különböztesse meg a szótárban tárolt, illetve nem tárolt szavak esetét!												
03.	BINKER.CPP	Valósítsa meg a bináris keresés algoritmusát iterációt és rekurziót használó függvények segítségével egyaránt!												
04.	BINTREE.CPP	Készítsen programot, amely bináris fa segítségével sorba rendezi a beolvasott számokat!												
05.	BIZTOSZTAS.CPP	Készítsen programot, amely kivételek alkalmazásával biztonságossá teszi két pozitív egész szám hányadosának számítását! Negatív operandusok esetén előjelet vált, kivédi a nullával való osztást, valamint kilép, ha az osztandó 0.												
06.	CUBE.CPP	Írjon egy-egy int , long és double típusú köbre emelő függvényt!												
07.	DETERMINANS.CPP	Készítsen determinánst számoló rekurzív függvényt!												
08.	DRENDLISTA	Alakítsuk át a neveket rendezett listában tároló példaprogram (RENDEZETTLISTA.CPP) adatstruktúráját kettős listává, ahol a második lista a névhez tartozó életkor szerint láncolja az elemeket!												
09.	EGYSEGM.CPP	Készítsen programot, amely egy 5x5-ös mátrix főátlójába 1, míg a többi elembe 0-át ír!												
10.	FELT1.CPP	Írjon programot a víz halmazállapotának vizsgálatára, amely beolvassa a víz hőmérsékletét, és a hőmérséklettől függően kiírja víz, jég vagy gőz szöveget!												
11.	FELT2.CPP	Készítsen programot az osztályzatok szerzett pontszámok alapján történő kiértékelésére. A program olvassa be a pontszámot, majd az alábbi táblázat alapján írja ki az osztályzatot: <table><tr><td>Pontszám</td><td>Osztályzat</td></tr><tr><td>< 40</td><td>elégtelen</td></tr><tr><td>40 .. 54</td><td>elégséges</td></tr><tr><td>55 .. 69</td><td>közepes</td></tr><tr><td>70 .. 84</td><td>jó</td></tr><tr><td>85 .. 100</td><td>jeles</td></tr></table>	Pontszám	Osztályzat	< 40	elégtelen	40 .. 54	elégséges	55 .. 69	közepes	70 .. 84	jó	85 .. 100	jeles
Pontszám	Osztályzat													
< 40	elégtelen													
40 .. 54	elégséges													
55 .. 69	közepes													
70 .. 84	jó													
85 .. 100	jeles													
12.	GYOK.CPP	Tanulmányozza a gyökvonás ciklussal való megoldását!												
13.	HAROMSZ.CPP	Olvassa be a háromszög három oldalának adatait és ellenőrizze a háromszög szerkeszthetőségét (bármely két oldal összegének nagyobbának kell lennie a harmadik oldalánál)! Írjon függvényeket a háromszög kerületének és területének kiszámítására! A terület kiszámítására használja fel a Heron-képletet!												
14.	JEGY.CPP	Írjon programot, amely a számként (1, 2, 3, 4, 5) beolvasott érdemjegyet szöveges formában jeleníti meg! Használja a switch utasítást!												

	<i>Fájlnev/könyvtár</i>	<i>Feladat</i>
15.	KARAKTERHALMAZ	Fejlesszen programot, karakterek halmazban történő kezelésére! Az elem-e információ tárolására karakterenként egy-egy bitet használjon!
16.	LINLISTA.CPP	Készítse el a 7.2.5 fejezetben bemutatott listakezelő program teljes változatát, függvények felhasználásával!
17.	MAGANH.CPP	Írjon programot, amely statisztikát készít a beolvasott karaktersorozatban található magánhangzóról!
18.	NEVSOR.CPP	Készítsen programot, amely a beolvasott (maximálisan 10 darab) nevet névsorba rendezi!
19.	QUICKSORT.CPP	Készítse el a gyorsrendezés (<i>quicksort</i>) algoritmusát megvalósító rekurzív függvényt!
20.	PRIM.CPP	Írjon olyan C++ függvényt, amely a paraméterként átadott n pozitív számról eldönti, hogy prímszám-e. A függvény visszatérési értékei logikai IGAZ prímszám esetén, illetve logikai HAMIS, ha a szám nem prímszám.
21.	RENDEZETTLISTA.CPP	Készítsen programot, amely a lineáris lista felépítésekor biztosítja a listában tárolt adatok (a legfeljebb 123-karakteres szövegek) rendezettségét!
22.	ROMAI.CPP	Írjon programot, amely 1-2000-ig kiírja a római számokat!
23.	ROMAISZ.CPP	Írjon függvényt, amely egy számot római számként ír ki!
24.	SAKKT.CPP	Készítsen egy sakktáblafeltöltő programot, amely a 8x8-as tömböt dinamikusan foglalja le, és a fekete kockákat 1-gyel, a fehéreket pedig 0-val tölti fel!
25.	SOR.CPP	Számítsa ki a $h = 1 + 1/2 + 1/3 \dots + 1/n$ sor összegét! Olvassa be az n értékét! Oldja meg a feladatot különböző ciklusutasításokkal külön függvényekkel!
26.	SZAMHALMAZ	Fejlesszen programot, egész számok halmazban történő kezelésére! Az elem-e információ tárolására számonként egy-egy bitet használjon!
27.	SZOROZ.CPP	Írjon programot, amely egy 10-elemű vektort feltölt egész számokkal, majd a páratlan indexű elemeket 7-tel, míg a páros indexű elemeket 12-vel megszorozza!
28.	TANKOR.CPP	Írjon programot, amely 12 elemű struktúrátömb típus segítségével az alábbi szerkezetű adatok tárolására használható! <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 45%;"> <p>adatok</p> <p>Név</p> <p>Ösztöndíj</p> <p>Tankör</p> <p>születési dátum:</p> </div> <div style="width: 45%;"> <p>típus</p> <p>char [30]</p> <p>float</p> <p>int</p> <p>struct datum</p> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div style="width: 45%;"> <p>év</p> <p>hó</p> <p>nap</p> </div> <div style="width: 45%;"> <p>int</p> <p>char[12]</p> <p>int</p> </div> </div> </div> </div>

	<i>Fájlnev/könyvtár</i>	<i>Feladat</i>
29.	<i>TANKOR2.CPP</i>	Egészítse ki a <i>TANKOR.CPP</i> programot úgy, hogy képes legyen a legnagyobb ösztöndíjjal rendelkező diák adatait megjeleníteni a képernyőn!
30.	<i>TOMB1.CPP</i>	Definiáljon egy 50-elemű tömböt, és adott elemszámra töltse fel értékekkel! Írjon beolvasó, kiíró függvényt, valamint számítsa ki az adatok átlagát, összegét, és keresse meg azokat az elemet, amelyik legjobban illetve legkevésbé tér el az átlagtól! Írjon olyan függvényt, amely megszámolja, hogy a tömbnek mennyi pozitív, negatív és zérus eleme van!
31.	<i>TOMB2.CPP</i>	Definiáljon egy 50-elemű tömböt, és adott elemszámra töltse fel értékekkel! Írjon függvényt, amely megszámolja, hogy mennyi páros és páratlan eleme van a tömbnek, valamint egy rendező függvényt, amely a tömb elemeit növekvő sorrendbe rendezi!
32.	<i>VAVEVI.CPP</i>	Írjon programot, amely beolvas egy mondatot! <i>Például Ma szep az ido.</i> <i>Visszaírja: MAVA SZEVEP AVAZ IVIDOVO.</i> Tehát a magánhangzó esetében v betűvel megismétli a magánhangzót: (i -> ivi, o -> ovo, e -> eve, a -> ava, u -> uvu).