

Adatok tömörítésének egyik módja a Huffman kódolás. Az októberben kitűzött **I. 276.** feladatban a kódolás menetét kellett egy prezentációban bemutatnunk, most feladatunk a kódolást végző program elkészítése.

A módszer lényege, hogy az eredeti állomány minden bájttját különböző hosszúságú bitsorozatra cseréli úgy, hogy a ritkábban előforduló bájtok hosszabb, a gyakoribb bájtok rövidebb bitsorozatot kapnak. A kód létrehozásának alapja, hogy az eredeti állomány egyes bájttjainak gyakorisága alapján létrehozunk egy bináris fát, majd megfelelő módon bejárjuk.

A fa felépítésének lépései a következők:

1. Először minden bájtot a gráf egy izolált pontjának tekintünk, amelynek nincsenek leágazásai, és értéke a benne szereplő karakter gyakorisága.
2. Ezután a két legkisebb gyakoriságú bájtt pontját összevonjuk, és befűzzük őket egy olyan új gráfpont bal és jobb ágába, amelynek értéke a két karakter gyakoriságának összege, majd a két eredeti gráfpontot töröljük.
3. A 2. pontot ismételjük addig, amíg az összes betűt föl nem fűztük egy bináris fába.

Az így létrejött fában a gyökérelemtől indulva megkapjuk az egyes karakterek Huffman kódját úgy, hogy a karakterhez vezető úton minden balra lépés 0-ás bitet és minden jobbra lépés 1-es bitet jelent.

Készítsük programot `s68` néven, amely a parancssor első argumentumaként megadott állomány bájttjainak egy lehetséges Huffman kódját előállítja, és a parancssor második argumentumaként megadott szöveges állományba írja soronként az egyes bájttokhoz rendelt új bitsorozatot és azok hosszát a kapott kód hossza, illetve azon belül ABC sorrendben a mintának megfelelő formában. Az ASCII tábla szerint nem nyomtatható karakterek és a szóköz helyett a karakter kódja jelenjen meg. A program ezen kívül írja a standard kimenetre a bemeneti állomány eredeti és kódolt méretét bit mértékegységben.

Bemeneti állomány	Kimeneti állomány	Standard kimenet
To be, or not to be: that is the question	32 = 01 (2) e = 001 (3) o = 100 (3) t = 110 (3) b = 0001 (4) h = 0000 (4) , = 10110 (5) : = 10101 (5) T = 10100 (5) i = 11111 (5) n = 11110 (5) s = 11100 (5) a = 111011 (6) q = 111010 (6) r = 101111 (6) u = 101110 (6)	Eredeti hossz: 328 bit Kódolt hossz: 148 bit

Beküldendő egy tömörített `s68.zip` állományban a program forráskódja (`s68.pas`, `s68.cpp`, ...) az `.exe` és más fordító által generált állományok nélkül, valamint a program rövid dokumentációja (`s68.txt`, `s68.pdf`, ...), amely tartalmazza a megoldás rövid leírását, és megadja, hogy a forrás melyik fejlesztő környezetben fordítható.