

Feladatok:

1. Készítsünk képernyővédő animációt, amely minden másodpercben a képernyőn véletlen helyre kiírja a dátumot és a pontos időt, majd egy másodperc után eltünteti azt. **(dos01)**

```
20XX.XX.XX. pentek
15:30:22:13
```

```
20XX.XX.XX. pentek
15:30:23:14
```

2. Írasd ki két billentyű lenyomása alatt eltelt időt! **(dos02)**

```
Nyomj le egy billentyut, majd meg
egyet!
Az eltelt ido: 85 szazadmasodperc.
```

3. Készítsünk programot, melyben generál egy 100 elmű tömbbe 0-1000 közötti számokat. Majd sorbarendezi valamelyik rendező algoritmussal. Utána pedig kimentí egy szamok.txt szövegfájlba. **(rend01)**

```
Rendezetlen: 453; 761; 33; ..
```

```
Rendezve: 12; 33; ...
```

4. A nev.txt text típusú állományban nevek vannak soronként (ha nincs ilyen fájl, akkor hozzá létre), írd át úgy, hogy rendezve legyenek! (max. 10 névvel) **.(rend02)**

```
Agoston Eszter
Horvath Eva
Horvath Zoltan
Kovacs Peter
...
Zelk Sandor
```

5. Készítsünk programot, amely rendezi egy rekord típusú tömbben levő diákok neveit növekvő sorrendbe. A tömb tartalmazza a diákok nevein kívül a születési évüket és születési helyüket is. **(rend03)**

```
1 ADATBEVITEL | 2 LISTAZAS | 0 KILEPES
Valassz: 2
Abert Tamas - 1998 - Szombathely
Andor Eva - 2001 - Budapest
Belso Peter - 2003 - Sarvar
...
Zente Zsolt - 1999 - Szombathely
```

19 Összefoglaló, fogalmak

• program nev;	{a program indítása, nevének megadása}
• var	{változók deklarálása}
• begin	{főprogram, alprogramrész kezdete}
• end; end.	{alprogram vége; főprogram vége}
• clrscr	{képernyőtörölés}
• write; writeln;	{kiíratás a képrnyőre; kiíratás a képernyőre sortöréssel}
• mod	{két egész szám osztásának maradéka}
• div	{két szám hányadosának egész része}
• read; readln;	{beolvasás billentyűzetről}
• integer	{egész típus}
• string	{szöveg}
• real	{valós szám}
• boolean	{igen; nem;}
• byte	{0-255 egész szám}
• char	{egy karakter}
• sqr()	{négyzetreemelés}
• sqrt()	{gyökvonás}
• for do	{ciklus}
• downto	{visszafelé számol}
• length()	{a változó hossza}
• upcase()	{nagybetűssé teszi}
• chr()	{egy karakter kiíratása az ASCII kód alapján}
• ord()	{egy karakter ASCII kódjának kiíratása}
• inc()	{függvényt (increase); egyel növeli az értéket}
• dec()	{egyel csökkenti az értéket}
• and; or; not; xor;	{logikai műveletek}
• if then	{ciklus}
• case of	{ciklus}
• while do	{előltesztelő ciklus}
• repeat until	{ciklus}
• randomize;	{véletlen szám generálásához meg kell adni a program elején}
• random()	{véletlen szám generálása}
• array of	{tömbök megadása}
• const	{állandók megadása}
• procedure	{eljárás}
• function	{függvény}

• delay()	{késleltetés ms –ban}
• gotoxy()	{pozícióba ugrás}
• keypressed	{boolean változó}
• readkey	{karakter beírására vár}
• textbackground	{háterszín megadása}
• textcolor	{betűszín megadása}
• window	{belső ablak rajzolása}
• sound(); nosound;	{hang be- és kikapcsolása}
• abs()	{abszolút érték}
• type	{típus megadása}
• record	{record típus}
• set of	{halmaz típus}
• text	{szöveges állományok deklarálása, text típusként}
• assign	{a parancs hozzárdel egy állományhoz egy fájl}
• reset	{megnyitja az állományt olvasásra}
• rewrite	{megnyitja az állományt írásra}
• halt	{befejeződik a program futása}
• eof	{end of file – a file végére}
• close	{bezárjuk az állományt}
• file	{típus nélküli állomány deklarálása}
• blockread	{típus nélküli állományból olvasás}
• blockwrite	{típus nélküli állományba írás}
• sizeof	{függvény – mennyi bájt van tárolva az adott változó}
• seek	{ugrás megadott fájlban belüli recordra ugrik}
• filepos	{függvény - megadja, hogy a direkt file hányadik rekordjánál tartunk}
• getdate	{eljárás – aktuális dátum lekérése}
• gettime	{eljárás – aktuális idő lekérése}