

A sok munkatárssal működő intézményekben költségtakarékoságból sok kis nyomtató helyett inkább néhány nagy kapacitását használják. Előnyt jelent a nyomtató nagyobb tudása, sebessége, alacsony üzemeltetési költsége. Egyetlen számottevő hátránya van: ha többen akarnak közel egy időben nyomtatni, akkor bizony várni kell. A papírkifogyás okozta idővesztésüket két, egyenlő méretű papíradagoló tálcával oldják meg. (A két tálcá között automatikusan, idővesztés nélkül vált a nyomtató, az üres tálcát azonnal újratöltik.)

A beérkező nyomtatási igényeket az egyik munkanap a `feladat.txt` állományban rögzítik a beérkezés sorrendjében. Az első sor első száma megadja, hogy a nyomtató egy lapot hány másodperc alatt nyomtat ki, a második szám pedig a papíradagoló tálcák külön-külön vett kapacitását jelöli. A további sorokban a nyomtatási feladat beérkezésének ideje (óra, perc, másodperc), a nyomtatandó oldalak száma és a nyomtatást kérő neve szerepel. Az oldalszám értéke legfeljebb 100 lehet. A cégnél legfeljebb kéttucatnyian dolgoznak.

Például:

```
feladat.txt
```

```
3 500
8 21 11 30 Anna
8 22 24 3 Eszter
8 22 26 10 Dominik
9 4 0 100 Anna
...
```

A 4. sor megmutatja, hogy a feladat Dominiktól 8:22:26-kor érkezett, amelyben 10 lap nyomtatását kéri. Ezen nyomtatási feladat végrehajtása 30 másodpercet igényel.

Készítsünk programot `nyomtat` néven, amely az alábbi kérdésekre válaszol. Ügyeljünk arra, hogy programunknak minden helyes tartalmú bemeneti állomány esetén működnie kell.

Minden részfeladat megoldása előtt írjuk a képernyőre a feladat sorszámát. Ha a felhasználótól kérünk be adatot, jelenítsük meg a képernyőn, hogy milyen értéket várunk (például a 2. feladat esetén: „2. feladat: Adja meg egy dolgozó nevét.”). A képernyőn megjelenített üzenetek esetén az ékezetmentes kiírás is elfogadott.

1. Olvassuk be a `feladat.txt` állományban talált adatokat, s azok felhasználásával oldjuk meg a következő feladatokat. Ha az állományt nem tudjuk beolvasni, az első 10 csomaghoz tartozó adatokat jegyezzük be a programba és dolgozzunk azzal.

2. Kérjük be a felhasználótól egy dolgozó nevét, majd írjuk a képernyőre, hogy az illető mikor küldött nyomtatási feladatot. Minden időpont új sorban jelenjen meg óra:perc:másodperc formában.

3. Készítsünk statisztikát, hogy az egyes órákban hány feladat érkezett. Az eredményt (az órát és a darabszámot) soronként írjuk a képernyőre. A nyomtatási feladatot nem tartalmazó órákat ne jelenítsük meg.

4. Adjuk meg azon dolgozók nevét, akik a vizsgált napon 9 óra előtt már nyomtattak. A neveket egymástól egy-egy szóközzel elválasztva, ábécé-rendben írjuk a képernyőre.

5. Listázzuk ki, mely nyomtatási feladatok megoldása közben váltott adagolót a nyomtató. A feladatok sorszámát egy-egy szóközzel elválasztva jelenítsük meg a monitoron.

6. Készítsünk kimutatást, amely feladatonként tartalmazza a feladat beérkezésének, a nyomtatás elkezdésének és a befejezésének idejét valamint a dolgozó nevét. Az adatokat soronként a `kesz.txt` állományba jegyezzük be.

```
kesz.txt
```

```
8 21 11 8 21 11 8 22 41 Anna
8 22 24 8 22 41 8 22 50 Eszter
8 22 26 8 22 50 8 23 20 Dominik
9 4 0 9 4 0 9 9 0 Anna
...
```

Beküldendő a program forráskódja (`i226.pas`, `i226.cpp`, ...), valamint a program rövid dokumentációja (`i226.txt`, `i226.pdf`, ...), amely tartalmazza a megoldás rövid leírását, és megadja, hogy a forrásállomány melyik fejlesztő környezetben fordítható egy tömörített `i226.zip` állományban.