

# Intel x86 Linux ASM - "Lööping"

**Kiértékelés:** Egy Intel 32 bit Linux assembly-ben írt **feladat.S** fájlt kell feltölteni. Ezt az assembly fájlt a bíró fordítja, szimbólumokat ellenőriz és végül linkeli. Ezek után futtatja és ellenőrzi, hogy **a kimenet megegyezik-e karakterre pontosan az elvárt eredménnyel.**

A bíró által végrehajtott fordítási és linkelési parancsok:

- Fordítás (+1 pont): `gcc -m32 -c -static <feladat>.S -o tmp.o`
- Linkelés (+1 pont): `gcc -m32 -static tmp.o -o program`

Futtatás (fordítás és linkelés után): `./program`

## Feladat (1 pont):

Egészítsük ki a **feladat** eljárásban lévő assembly kezdeményt egy ciklus megadásával. Csak a kijelölt helyre írjunk!

A feladat:

- Járjuk be azt az integer tömböt aminek a kezdő címét az **esi** regiszter tartalmazza.
- A tömb elemeinek a számát az **edx** regiszter tartalmazza (azaz azt, hogy hány darab integert kell feldolgoznunk).
- Azokat a tömb elemeket melyek 2-nél nagyobbak vagy egyenlőek másoljuk egymás után a kimeneti tömbbe.
- A kimeneti tömb címét a **edi** regiszter már tartalmazza.
- Eközben az **eax** regiszterbe számoljuk, hogy hány elemet másoltunk a kimeneti tömbbe. A ciklus végére ennek az **eax** regiszternek kell tartalmaznia hány elem került összesen a kimeneti tömbbe.

Megjegyzések:

- Az ellenőrzés során a kimeneti tömb tartalma és az "eax" regiszter is vizsgálatra kerül.
- A bemeneti és kimeneti tömbök előjeles egész értékeket tárolnak (32 bit integer).
- A bemeneti tömb elemeit ne módosítsuk, mert futási hibához vezethet.
- Nem kell eljárást csinálni!
- A kimeneti tömb alapértelmezetten a **0xFF** értékkel van feltöltve.
- Ne írjuk túl a kimeneti tömböt (vizsgálva van).
- Kettő teszt eset van előre megadva.
- Egy darab **.S** kiterjesztésű Intel 32 bit assembly-t tartalmzó fájlt kell feltölteni.