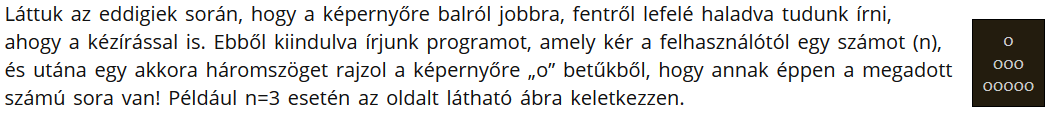
# 2. labor

Tematika:

* MÁSODIK CSOPORTÁL EMLÍTENI: gcc -lm math.h esetén és sizeof!!!
* if-else (írjuk ki két szám közül a nagyobbat)
* egymásba ágyazott if-else-ek (másodfokú egyenlet probléma)
  + 3, -5, -2
  + 1, 6, 5
  + 1, -8, 25
  + 1, -14, 49
* switch-case: osztályzat
* For ciklusok, megszámlálási szerep: legnagyobb közös osztó probléma, naív módszer
* While ciklus, végtelen, break kulcsszó
  + while(1){ if(a>b) a-=b; else if (a<b) b-=a; else break;}
  + while(a!=b) {if(a>b) a-=b; else b-=a;}
  + while(1){r=a%b; if(r==0) break; ‘else’ {a=b; b=r;} } printf(b);
  + r=a%b; (!!) while(r!=0){a=b; b=r; r=a%b;} printf(b);
* A for nem csak megszámláló:
  + for(r=a%b; r!=0; r=a%b){a=b; b=r;}
  + for(;r=a%b!=0;){a=b;b=r;}
  + for(;r=a%b;){a=b;b=r;}
  + for(scanf(a,b); r=a%b; a=b, b=r);
* continue: írjuk ki 1-100-ig a számokat, kivéve az 5-tel oszthatókat
* Olvasás EOF-ig

Órai feladatok:

* <https://progcont.hu/progcont/100194/?pid=201020>
* 
* <https://progcont.hu/progcont/100198/?pid=201036>

Házi feladatok:

* Írjunk programot, amely 1-3000 közötti természetes számokat vált át római számmá! Tetszőleges számú természetes számot olvashatunk be, a bemenet végét EOF jelzi! A megoldás ne tartalmazzon felhasználói üzeneteket! + RANDOM!
* Olvassunk be pozitív egész számokat, majd állapítsuk meg, hogy az adott számok tükörszámok-e (egy szám tükörszám, ha a szám ugyanaz balról jobbra, mint jobbról balra olvasva, pl. 1331, 75957). A legutolsó beolvasott szám a 0 (ezt már nem kell feldolgozni). A megoldás ne tartalmazzon felhasználói üzeneteket!
* <https://progcont.hu/progcont/100245/?pid=201202>