

1. feladat: Milyen bemenet kell? (50 pont)

A lenti programnak egy számsorozatot úgy kell kiírnia, hogy először a benne szereplő negatív elemeket írja, majd pedig a nemnegatívakat!

A. Melyek a bemenő adatok?

B. Milyennek kell lenniük, hogy a következő programrészlet jól működjön?

C. Milyen sorrendben írja ki az algoritmus a negatív, illetve a pozitív számokat!

D. Fogalmazd meg, hogy mit jelent $B[i]$ értéke?

```
F1:=0; F2:=0; x:=0
Ciklus i=1-től N-ig
    Ha A[i]≥0 akkor Ha F2>0 akkor B[F2]:=i különben y:=i
                    F2:=i
                    különben Ha F1>0 akkor B[F1]:=i különben x:=i
                    F1:=i
Ciklus vége
B[F1]:=y; B[F2]:=0
Ciklus amíg x≠0
    Ki: A[x]; x:=B[x]
Ciklus vége
```

2. feladat: Veremautomata (56 pont)

Egy veremautomata a következőképpen működik:

- ha számot kap a bemenetén, azt a verembe teszi,
- ha műveleti jelet kap, akkor a verem tetején levő egy vagy két számot kiveszi, velük a műveletet végrehajtja, majd az eredményt a verembe teszi.

A veremautomata bemenetén a következő sorozatot kapja:

1 3 + 2 5 4 * 8 - 6 / + négyzetgyök -

Példa: 1 3 - esetén kiveszi a veremből a 3-at és az 1-et, kiszámítja $1-3=-2$ -t, majd beteszi a verembe a -2 -t.

Add meg, mi lesz a verem állapota az egyes jelek (adat vagy művelet) érkezése után, illetve a feldolgozás végén!

3. feladat: Mit csinál (44 pont)

A. Fogalmazd meg, mik a hibák a következő programban, ami egy 1-nél nagyobb természetes szám valódi osztói számát (azaz nem beleértve az 1-et és önmagát) és a legkisebb valódi osztóját írja ki (de csak ha van)!

B. Mi a feltétele a helyes működésnek (mindkét részfeladatban)?

```
Be: N; K:=0; L:=0
Ciklus i=2-től egészrész (gyök(N))-ig
    Ha N mod i=0 akkor K:=K+1
    Ha K=1 akkor L:=i
Ciklus vége
Ki: K*2
Ki: L
```

4. feladat: Titkosítás (54 pont)

Számítógéppel titkosítottak egy értelmes magyar SZÓ-t. A program lefutása után közölték velünk az előállított KÓD-ot, de a használt egybetűs KULCS-ot nem.

A kódoláskor használt algoritmus a következő volt:

```
Kódolás:
KÓD[0] := KULCS
Ciklus I=1-től HOSSZ-ig
    KÓD[I] := SZÓ[I] művelet KÓD[I-1]
Ciklus vége
Eljárás vége.
```

Az 1-től HOSSZ-ig indexelt SZÓ tömb a kódolandó, a 0-tól HOSSZ-ig indexelt KÓD tömb pedig a kódolt szót tartalmazza.

A KÓD (1-től HOSSZ-ig) ez lett: HIKJFG. Az alábbi igazságtáblával megadott műveletet a számítógép az operandusokon bitenként végzi el (A és B az operandusok, C az eredmény).

Művelet:

A	0	1	0	1
B	0	0	1	1
C	0	1	1	0

A használható betűk és kódjaik:

- 0000	A 0001	B 0010	C 0011
D 0100	E 0101	F 0110	G 0111
H 1000	I 1001	J 1010	K 1011
L 1100	M 1101	N 1110	O 1111

Mi volt az eredeti, értelmes magyar szó (megjegyzés: az első betű kitalálásához az algoritmus nem ad segítséget), és mi volt a KULCS? Magyarázd meg, hogyan jöttél rá a megoldásra!

5. feladat: Assembly (40 pont)

Az assembly nyelv a számítógép processzora által futtatott egyszerű gépi utasítások szöveges formája. Ciklusokat és elágazásokat a kód megfelelő részére – melyet egy szöveges címke jelöl – való feltételes ugrással képezhetünk benne. Változókra nem névvel utalunk, hanem az alapján, hogy a processzor melyik regiszterében (belső tárolóegység) található.

Az alábbi program az A és B memóriacímen lévő, 64 bites nemnegatív egész számokból számol ki egy 64 bites, nemnegatív egész értéket, majd azt a C címen tárolja. A segédszámításokhoz hat 64 bites regisztert (x0-x5) használ.

```
ldr    x0, A           ; az x0 regiszterbe az A címen található értéket betölti
ldr    x1, B           ; az x1 regiszterbe a B címen található értéket betölti
mov    x2, 0           ; az x2 regiszterbe 0-t tölt
mov    x3, 0           ; az x3 regiszterbe 0-t tölt
mov    x4, 64          ; az x4 regiszterbe 64-et tölt
```

X:

```
lsl    x2, x2, 1       ; az x2 regiszter tartalmát eggyel balra lépteti
lsl    x3, x3, 1       ; az x3 regiszter tartalmát eggyel balra lépteti
lsr    x5, x0, 63      ; az x5 regiszterbe x0 értékének legnagyobb helyiértékű bitjét tölti
orr    x3, x3, x5      ; az x3 regiszterbe az x3 és x5 közti bitenkénti VAGY
```

```

                                ; eredményét tölti
cmp    x3, x1                    ; x3 és x1 értékét összehasonlítja
blo   Y                          ; az Y címkére ugrik, ha x3 < x1
sub   x3, x3, x1                 ; az x3 regiszter értékéből kivonja x1 regiszter értékét
add   x2, x2, 1                  ; az x2 regiszter értékéhez 1-et ad
Y:
lsl   x0, x0, 1                  ; az x0 regiszter tartalmát eggyel balra lépteti
subs  x4, x4, 1                  ; az x4 regiszter értékéből 1-et kivon
bne   X                          ; az X címkére ugrik, ha a kivonás eredménye nem 0 volt
str   x2, C                      ; a C memóriacímen tárolja x2 értékét

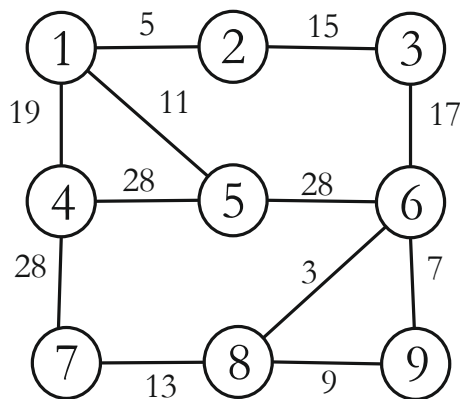
```

Kiegészítés: egy kettes számrendszerbeli szám eggyel balra léptetése azt jelenti, hogy a legmagasabb helyiértékű bitet töröljük, az végére pedig 0-t teszünk, például, 4 biten: $1011_b \ll 1 == 0110_b$.

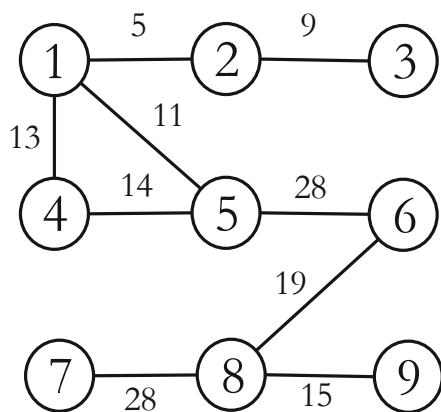
- Mi lesz a C címen és az x3 regiszterben, ha $A=10, B=3$?
- Mi lesz a C címen és az x3 regiszterben, ha $A=18, B=3$?
- Mi lesz a C címen és az x3 regiszterben, ha $A=26, B=1$?
- Mi lesz a C címen és az x3 regiszterben, ha $A=34, B=7$?
- Hogy függ a C címen található érték A-tól és B-től? Írd le általánosan!
- Mi lesz az x3 regiszter tartalma a futás végén?

6. feladat: Utak (80 pont)

Bergengóciában a városokat földutak kötik össze. A király elhatározza, hogy az utakat leaszfaltozzák, de sajnos nincs annyi pénze, hogy minden utat aszfaltozzanak. A királynak K forintja van. A következő stratégiát választotta: Először aszfaltozzák a legkevesebbe kerülő utat. Ezután minden lépésben kiválasztják a legolcsóbb olyan utat, ami olyan városokat köt össze, amelyek egyikéből a másik nem érhető el aszfaltozott úton. Add meg az alábbi két városhálózatra, hogy ezzel a szabállyal mely utak lesznek leaszfaltozva, az aszfaltozás sorrendjében és a királynak mennyi pénze marad!



- A királynak 30 forintja van.
 - A királynak 60 forintja van.



B.

B1. A királynak 40 forintja van.

B2. A királynak 80 forintja van.

7. feladat: Siklóernyő (80 pont)

Alex siklóernyőzni szeretne a Tátra hegycsúcsain. A hegycsúcsok a Tátrában egy egyenes mentén helyezkednek el és mindegyiknek ismert a magassága. A siklóernyőzés során Alex elindul az egyik kiválasztott hegycsúcsról és átrepül valamelyik (nem feltétlenül szomszédos), a jelenleginél szigorúan kisebb magasságú hegycsúcsra (akkor is, ha egy másik repülésnél már járt ott). Egy csúcsról akkor is átmehet egy alacsonyabb hegycsúcsra, ha a kettő között van magasabb hegycsúcs – ezeket a levegőben kikerüli.

Alex a siklóernyőzés során több alkalommal is átrepülhet a jelenlegi csúcsról egy következőre, és bármelyiken be is fejezheti a siklóernyőzést, feltéve, hogy legalább egy átrepülést végrehajtott már.

Add meg **minden egyes hegycsúcsra** a lentebb megadott magasságok esetén, hogy onnan kiindulva

A. hány különböző siklóernyőzést tehet, mely során pontosan egy alkalommal repül át másik csúcsra?

B. hány különböző siklóernyőzést tehet, mely során pontosan két alkalommal repül át másik csúcsra?

C. hány különböző siklóernyőzést tehet összesen?

7	8	7	9	3	2	4	3	3	6
---	---	---	---	---	---	---	---	---	---