

AZTERKETA

Informatikaren 0inarriak

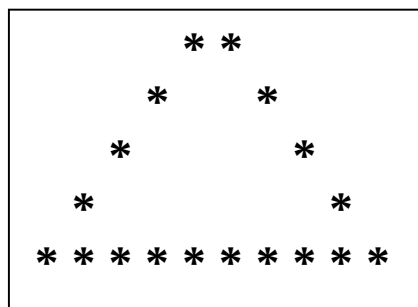
Azken Nota: Azterketa Teorikoa * 0.20 + Azterketa Praktikoa * 0.80

Nota: Formula hau aplikatu ahal izateko biak gainditu behar dira

1. (1 puntu) Zer idatziko du pantailan ondorengo programak? (sarrera datu bezala 5 balioa izanik)

```
DIM alt, a, z AS INTEGER
```

```
INPUT alt
FOR a = 1 TO alt - 1
  FOR z = a+1 TO alt
    PRINT " ";
  NEXT z
  PRINT "*";
  FOR z = 1 TO (a - 1) * 2
    PRINT " ";
  NEXT z
  PRINT "*"
NEXT a
FOR a = 1 TO alt * 2
  PRINT "*";
NEXT a
```



2. (1 puntu) Ondorengo programak zenbaki baten faktoriala kalkulatzen du, zuzendu egon daitezkeen errore posibleak:

```
DIM iZenb, iFaktoriala, iKont AS INTEGER
INPUT "Sartu zenbaki oso bat "; iZenb
iFaktoriala = 1
FOR iKont=iZenb TO 0 2 STEP -1 (edo FOR iKont = 2 to iZenb)
  iFaktoriala = iFaktoriala * iKont
END FOR NEXT iKont
PRINT iFaktoriala iZenb;" zenbakiaren faktoriala";iZenb iFaktoriala;" da."
```

3. (3 puntu) N zenbaki oso bat emanda, mezu baten bidez ea zenbaki lehena den edo ez esaten duen algoritmoa espezifikatu eta idatzi (1,5 puntu). Baita ere, programa idatzi Qb lengoaia erabiliz (1,5 puntu)

algoritmoa Lehena_Da

hasiera

```
Irakurri (N)
Kontagailua = 0
egin I guztietarako 1 tik N raino
  baldin N mod I = 0 orduan
    Kontagailua = Kontagailua + 1
  ambaldin
amguztietarako
  baldin Kontagailua = 2 orduan
    idatzi_Mezua(N, " lehena da")
  bestela
    idatzi_Mezua(N, " ez da lehena")
  ambaldin
amaia.
```

```
DIM N, I, Kontagailua as INTEGER
```

```
Irakurri (N)
Kontagailua = 0
FOR I = 1 TO N
  IF N mod I = 0 THEN
    Kontagailua = Kontagailua + 1
  END IF
NEXT I
IF Kontagailua = 2 THEN
  PRINT N; " lehena da"
ELSE
  PRINT N; " ez da lehena"
END IF
```

Modu eraginkorrago batez zenbakia bera eta 1 ea zatitzaileak diren aztertu gabe:

```

algoritmoa Lehenada
  hasiera
    Irakurri (N)
    Kontagailua = 0
    egin I guztietarako 2 tik N - 1 raino
      baldin N mod I = 0 orduan
        Kontagailua := Kontagailua + 1
      ambaldin
    amguztietarako
    baldin Kontagailua = 0 orduan
      idatzi_Mezua(N, "lehena da")
    bestela
      idatzi_Mezua(N, "ez da lehena")
    ambaldin
  amaia.

```

```

DIM N, I, Kontagailua as INTEGER

Irakurri (N)
Kontagailua = 0
FOR I = 2 TO N - 1
  IF N mod I = 0 THEN
    Kontagailua = Kontagailua + 1
  END IF
NEXT I
IF Kontagailua = 0 THEN
  PRINT N; "lehena da"
ELSE
  PRINT N; "ez da lehena"
END IF

```

Edo are eraginkorragoa, 1 edo zenbaki bera ez den lehenengo zatitzailea aurkitzen denean geldituz:

```

algoritmoa Lehenada
  hasiera
    Irakurri (N)
    B = egiazkoa
    I = 2
    bitartean B and I <= N - 1 egin
      baldin N mod I = 0 orduan
        B = faltsua
      ambaldin
    ambitartean
    baldin B orduan
      idatzi_Mezua(N, "lehena da")
    bestela
      idatzi_Mezua(N, "ez da lehena")
    ambaldin
  amaia

```

```

DIM N, I, B as INTEGER
Const EGIAZKOA=1, FALTSUA=0

Irakurri (N)
B = EGIAZKOA
I = 2
WHILE B and I <= N - 1
  IF N mod I = 0 THEN
    B = FALTSUA
  END IF
WEND
IF B THEN
  PRINT N; "lehena da"
ELSE
  PRINT N; "ez da lehena"
END IF

```

Errepikatzen segitzeko baldintzan ($I \leq N-1$) adierazpenaren ordeztuz
 ($I \leq N \bmod 2 + 1$) erabil liteke N hiru baino handiago bada, horrela eraginkortasuna handiagotuz

4. 4. (5 puntu) Banku batean 100 kontu korrante administratzeko gaitasuna edukiko duen programa bat nahi dute. Kontu zenbakiak 1-tik 100-ra kodetuta daude. Hasieran kontuetan ez dago dirurik, hau da, kontuen saldoa 0 da.

Leihatilako langileak 4 eragiketa mota egin ditzake:

- (1) diru-sarrera: Emandako diru-kantitatea dagokion kontu-zenbakian gehitu.
- (2) diru-irteera: Emandako diru-kantitatea dagokion kontu-zenbakitik kendu.
- (3) diru-transferentzia: Emandako diru-kantitatea lehenengo kontutik kendu eta bigarrenko kontuan gehitu.
- (0) programaren amaiera.

Lehenengo 2 eragiketarako *eragiketa-kodea*, *kontu zenbakia* eta *diru-kantitatea* dira emango diren datuak. Aldiz, diru-transferentzia egiteko *eragiketa-kodea*, *kontu zenbaki 1*, *kontu zenbaki 2* eta *diru-kantitatea* dira beharrezko diren datuak.

Posible da kontu-korronteko saldoa negatiboan edukitzea. Beraz, nahiz eta diru-irteera edo -transferentzia egiteko eskatzen zaigun diru-kantitatearik ez eduki kontu-korrontean, eragiketa egin daiteke.

Programak ondorengoa egin behar du:

- Eskatzen diren eragiketak burutu, eragiketa-kodea 0 izan arte
- Idatzi egunean zehar langileak egin dituen eragiketa ezberdinen kopurua portzentaiaetan
- Idatzi kontu guztiek duten saldoaren egoera
- Idatzi zein kontuk duen saldo handiena eta zenbatekoa den

Adibidea:

Eragiketak:	Kodea	Kontu-Zbki1	Kontu-Zbki2	Kantitatea
	1	3		1000
	1	5		500
	1	2		500
	2	3		300
	2	5		1000
	1	4		300
	3	2	6	200
	3	4	100	100
	0			

Emaitzak:

Eragiketa kopurua: %50 diru-sarrerak, %25 diru-irteerak eta %25 diru-transferentziak

Kontuen egoera:	Kodea	Saldoa
	1	0
	2	300
	3	700
	4	200
	5	-500
	6	200
		
	100	100

saldo handiena 3 kontu-zenbakiak du eta 700-ekoa da

```
CONST KontuMax = 100
DIM eragKodea, kZenb1, kZenb2, kantitatea AS INTEGER
DIM kontuak(1 TO KontuMax) AS INTEGER
DIM eragKop, erag1Kop, erag2Kop, erag3Kop AS INTEGER
DIM p1, p2, p3 AS SINGLE
DIM kzMax AS INTEGER

'Menua erakutsi eta esandako eragiketa egin
DO
  CLS
  PRINT "Eragiketen menua:"
  PRINT " (1) diru-sarrera"
  PRINT " (2) diru-irteera"
  PRINT " (3) diru-trasferentzia"
  PRINT " (0) programaren amaiera"
  PRINT
  PRINT " Zer egin nahi duzu:";
  INPUT eragKodea

  SELECT CASE eragKodea
    CASE 1: INPUT " Kontu zenbakia:", kZenb1
      INPUT " Kantitatea: ", kantitatea
      kontuak(kZenb1) = kontuak(kZenb1) + kantitatea
      erag1Kop = erag1Kop + 1
    CASE 2: INPUT " Kontu zenbakia:", kZenb1
      INPUT " Kantitatea: ", kantitatea
      kontuak(kZenb1) = kontuak(kZenb1) - kantitatea
      erag2Kop = erag2Kop + 1
    CASE 3: INPUT " 1go Kontu zenbakia:", kZenb1
      INPUT " 2. Kontu zenbakia:", kZenb2
      INPUT " Kantitatea: ", kantitatea
      kontuak(kZenb1) = kontuak(kZenb1) - kantitatea
      kontuak(kZenb2) = kontuak(kZenb2) + kantitatea
      erag3Kop = erag3Kop + 1
    CASE 0:
    CASE ELSE: PRINT "Emandako zenbakia gaizki dago. Aukeratu zuzenekoa"
  END SELECT
LOOP UNTIL eragKodea = 0

'Idatzi egindako eragiketen portzentaiak
eragKop = erag1Kop + erag2Kop + erag3Kop
p1 = erag1Kop / eragKop * 100
p2 = erag2Kop / eragKop * 100
p3 = erag3Kop / eragKop * 100
PRINT "Eragiketa kopurua: "
PRINT " %"; p1; " diru-sarrerak"
PRINT " %"; p2; " diru-irteerak"
PRINT " %"; p3; " diru-transferentziak"

'Idatzi kontuen egoera
PRINT "Kontuen egoera: "
FOR kZenb1 = 1 TO KontuMax
  PRINT " "; kZenb1; " "; kontuak(kZenb1)
NEXT kZenb1
```

'Idatzi saldo haundiena duen kontu-zenbakia eta kantitatea

kzMax = 1

FOR kZenb1 = 2 **TO** KontuMax

IF kontuak(kZenb1) > kontuak(kzMax) **THEN**

 kzMax = kZenb1

END IF

NEXT kZenb1

PRINT "Saldo handiena "; kzMax; " kontu-zenbakiak du eta "; kontuak(kzMax); "-ekoa da."

Programa idazteko beste aukera bat: lehenengo zatian baldintzazko sententziak erabiliz, aukera anitzekoaren orde:

'Menua erakutsi eta esandako eragiketa egin

DO

 CLS

PRINT "Eragiketen menua:"

PRINT " (1) diru-sarrera"

PRINT " (2) diru-irteera"

PRINT " (3) diru-trasferentzia"

PRINT " (0) programaren amaiera"

PRINT

PRINT " Zer egin nahi duzu:";

INPUT eragKodea

IF eragKodea = 1 **THEN**

INPUT " Kontu zenbakia:", kZenb1

INPUT " Kantitatea: ", kantitatea

 kontuak(kZenb1) = kontuak(kZenb1) + kantitatea

 erag1Kop = erag1Kop + 1

ELSEIF eragKodea = 2 **THEN**

INPUT " Kontu zenbakia:", kZenb1

INPUT " Kantitatea: ", kantitatea

 kontuak(kZenb1) = kontuak(kZenb1) - kantitatea

 erag2Kop = erag2Kop + 1

ELSEIF eragKodea = 3 **THEN**

INPUT " 1go Kontu zenbakia:", kZenb1

INPUT " 2. Kontu zenbakia:", kZenb2

INPUT " Kantitatea: ", kantitatea

 kontuak(kZenb1) = kontuak(kZenb1) - kantitatea

 kontuak(kZenb2) = kontuak(kZenb2) + kantitatea

 erag3Kop = erag3Kop + 1

ELSEIF eragKodea <> 0 **THEN**

PRINT "Emandako zenbakia gaizki dago. Aukeratu zuzenekoa"

END IF

LOOP UNTIL eragKodea = 0