# Algoritmus

Definíció: "Az algoritmus a feladat megoldására szolgáló, egyértelmű módon, és sorrendben végrehajtott utasítások sorozata, amely véges időn belül befejeződik"

* Beolvasás (adatok, melyekkel az algoritmus műveletet végez)  
  Folyamatábra:   
    
    
  Struktogram:  
    
    
    
    
  Szövegalapú: „be”

Be:

Be:

* Kiíratás (Eredmények, részeredmények közlése a felhasználóval)  
  Folyamatábra:   
    
    
  Struktogram:   
    
    
    
  Mondatszerű: „Ki:”

Ki:

Ki:

* Értékadás (bal oldalon levő változó értéket kap, a jobb oldalán szereplő kifejezés segítségével)  
  A kifejezésben szerepelhet változó vagy konstans, ill. ezek összekapcsolása valamilyen operátorral  
  A változó az adat pillanatnyi értékét adja meg, a konstans pedig olyan hivatkozás, amely időben nem változtatja meg az értékét  
  Folyamatábra:   
    
  Struktogram:  
    
    
  Mondatszerű: „v:=”

v:=

v:=

* Elágazás (Logikai feltételtől függ, hogy az algoritmus melyik irányba folytatódik tovább, logikai feltételnek azt a feltételt nevezzük, amelyről egyértelműen eldönthető, hogy igaz vagy hamis)  
  Folyamatábra:

log.felt.

* Struktogram:  
    
    
    
    
    
    
    
  Mondatszerű: Ha (log.felt.) akkor alg1, különben alg2

log. felt.

alg. 2

alg.1

* Többirányú elágazás (adott N darab feltétel, és mindegyikhez tartozik egy utasítássorozat, attól függően, hogy melyik feltétel teljesül, a hozzá tartozó utasítássorozat kerül végrehajtásra)
* Ciklus (amennyiben egy algoritmus részletet többször szeretnénk megismételtetni, akkor azt ciklussal valósítjuk meg), 2 részből áll
  + Ciklusfeltétel (Felügyeli, hogy hányszor fusson le a ciklusmag)
  + Ciklusmag (-ba azokat az utasításokat szervezzük, amelyeket ismételtetni szeretnénk)  
    3 ciklust különböztetünk meg:
  + Számlálóciklus (akkor alkalmazzuk, ha pontosan tudjuk, hogy hányszor kell lefutnia a ciklusmagnak)  
    Struktogram:  
      
      
      
      
    CV = Ciklusváltozó  
    ké = kezdőérték  
    vé = változóérték  
    lk = lépésköz  
    CM = Ciklusmag  
      
    Mondatszerű:  
    Ciklus CV:= ké->vé lk  
    CM  
    Ciklus vége

CM

CV=ké->vé lk

* + Előltesztelő ciklus (előbb teszteli a feltételt, és ha a feltétel igaz, akkor végrehajtja a Ciklusmagot, üres ciklus akkor alakul ki, ha a feltétel az első tesztnél hamis, végtelen ciklus pedig akkor, ha a feltétel sosem válik hamissá)  
    Struktogram:  
      
      
      
      
      
    Mondatszerű:  
    Ciklus amíg (log.felt)  
    CM  
    Ciklus vége

CM

CF (ciklusfeltétel)

* + Hátultesztelő ciklus (Előbb lefut a CM, utána vizsgálja a feltételt, a CM legalább egyszer lefut)  
    Struktogram:  
      
      
      
      
      
    Mondatszerű:  
    Ciklus   
    CM  
    amíg (log.felt.)

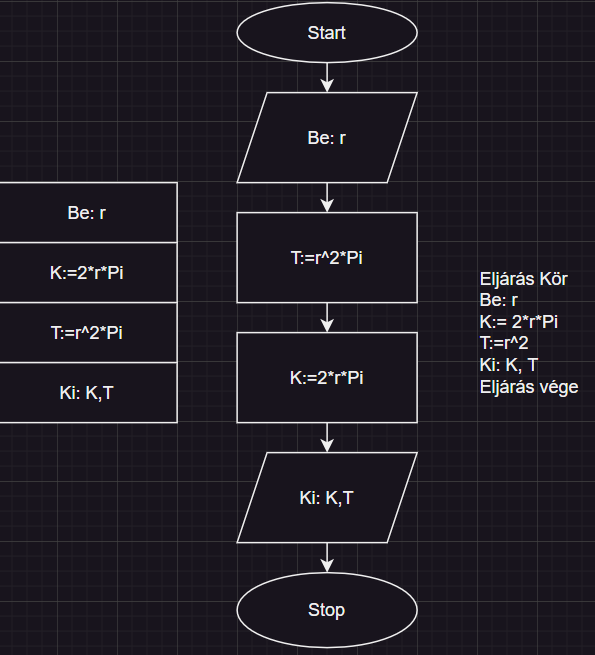
CM

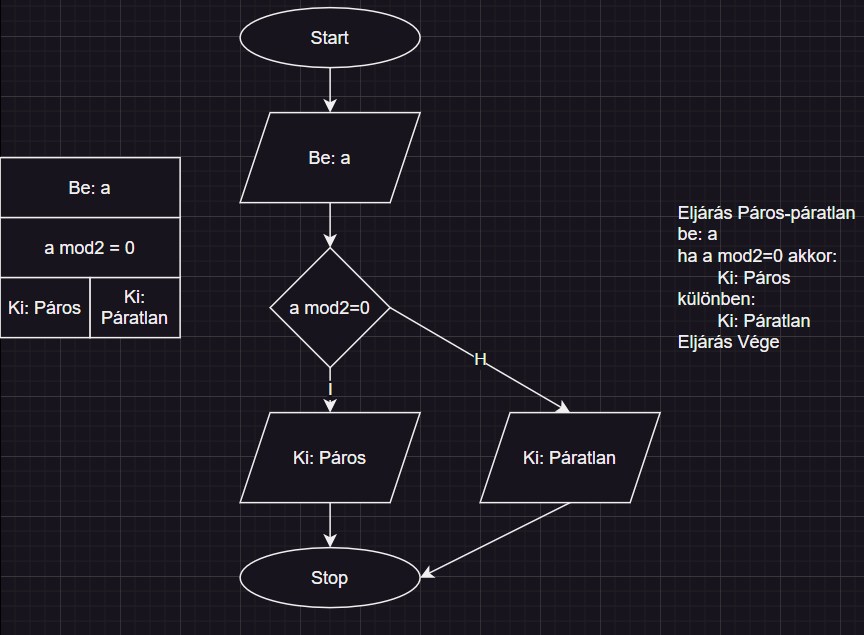
CF

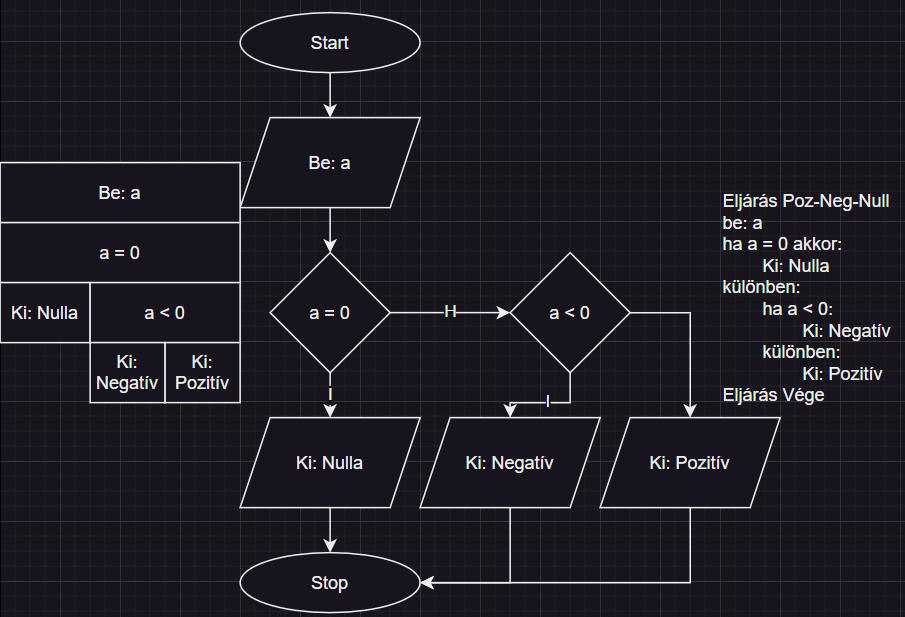
Írj algoritmust, mely meghatározza, a téglalap kerületét, területét

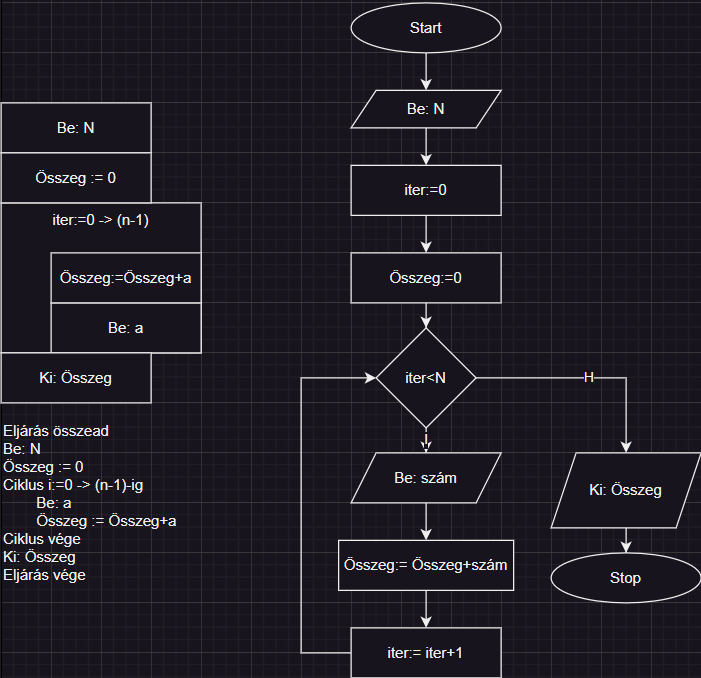
A képen szöveg, képernyőkép, diagram, tervezés látható

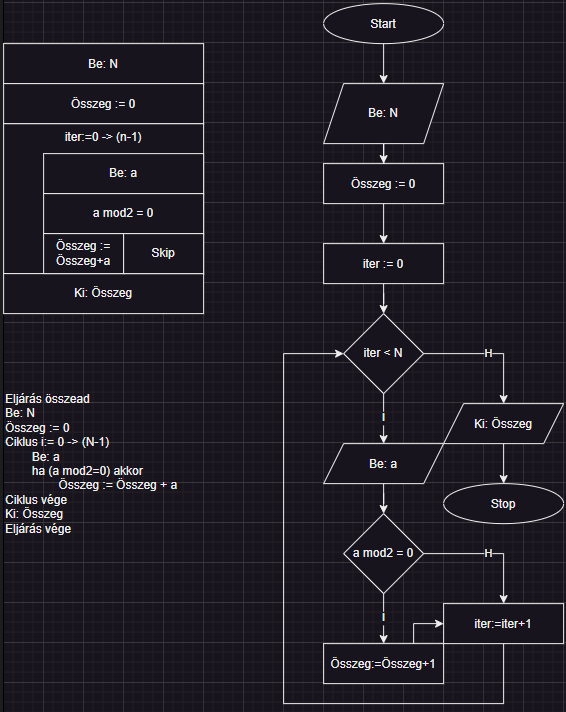
Automatikusan generált leírás

amely meghatározza a kör ker-,ter-ét  


amely eldönti egy számról, hogy páros vagy paratlan  


amely eldönti egy számról, hogy pozitív, negatív, vagy nulla  


amely összead N db számot  


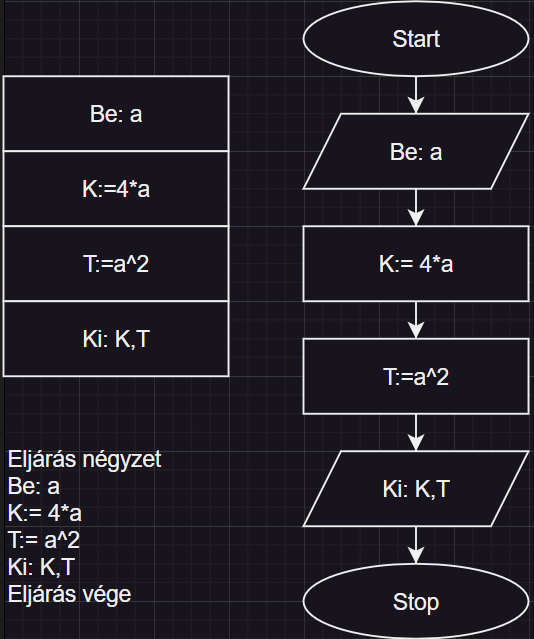
adott egy N elemű sorozat, amely meghatározza a párosok összegét  


Amely összead N db szám van ,csak pozitívakat összeadni  
A képen szöveg, diagram, képernyőkép, tervezés látható

Automatikusan generált leírás

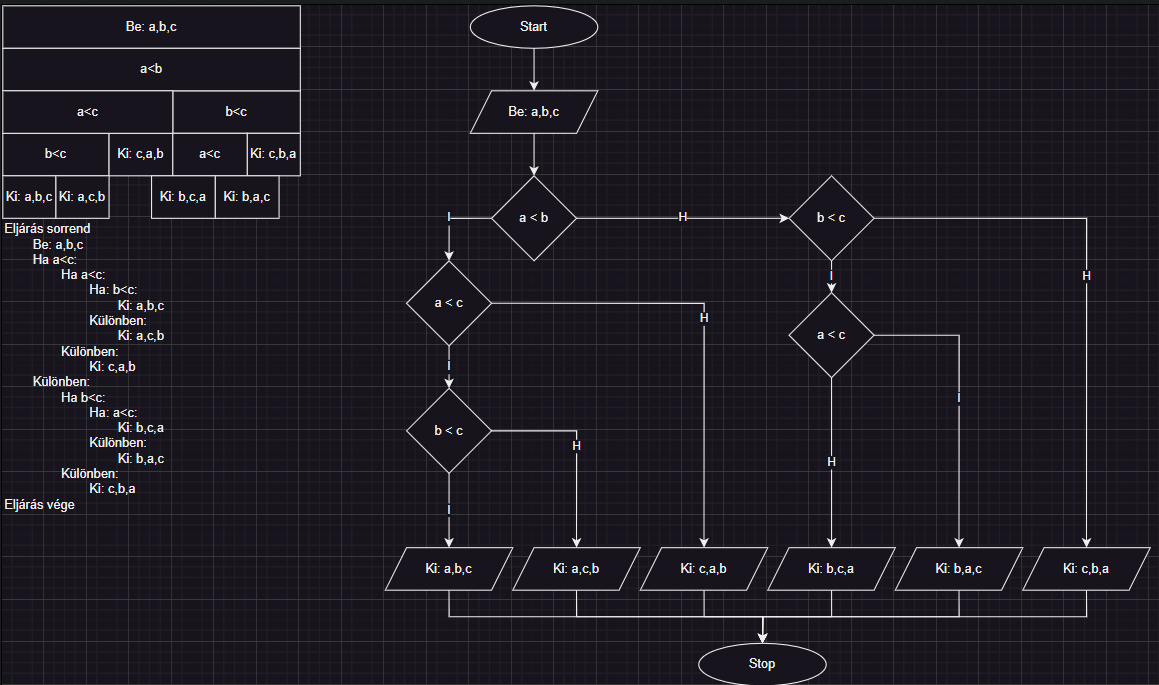
Ha valaki kérdi, ez Draw.io

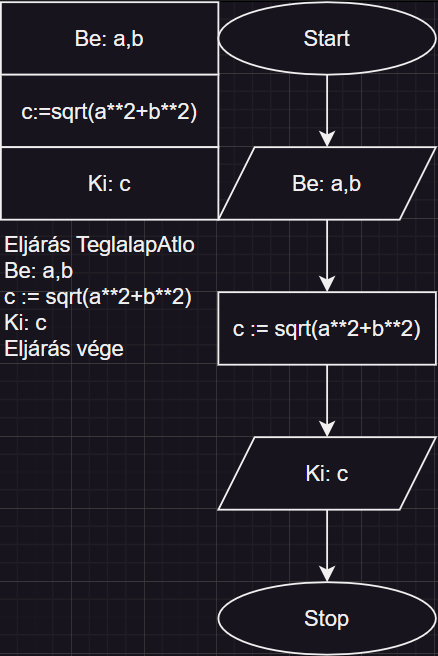
Négyzet ker-,területe



Pitagorasz algoritmus:  
A képen szöveg, képernyőkép, diagram, Betűtípus látható

Automatikusan generált leírás

3 beolvasott egész számot növekvő sorrendbe tesz  


algoritmus, mely egy téglalapnak az oldalai ismeretével kiszámítja az átlóját  


mely két beolvasott szám közötti abszolút eltérést megadja

A képen szöveg, képernyőkép, diagram, tervezés látható

Automatikusan generált leírás

melynél adott 3 egész szám, megállapítja, hogy alkothatják-e egy háromszög oldalait, amennyiben igen, határozza meg kerületét, területét  
A képen szöveg, képernyőkép, diagram, tervezés látható

Automatikusan generált leírás

2 egész szám, viszony (kisebb-nagyobb-egyenlő)

# A Tömb adatszerkezet

* A jobb neve: Array
* Jellemzői:
  + Homogén (Csak egy típusú adat lehet benne)
  + Statikus (Mérete inicializálásnál megadódik, mérete nem változtatható a program futása alatt)
  + Indexált hozzáférés (0-ról indul (N-1)-ig, és például array[N+1] == array[0] lesz, ha az index N-en túl megy, visszamegy 0-ig, így például array[N+5] == array[4])
  + Egyszerű adattípusokkal való munka (Akármelyik egyszerű típust el tud tárolni (int, double, string, bool, stb.))
* Deklarálására példák:
  + „int[] szam = new int[5]” (5 darab int-re való helyet lefoglal)
  + „string[] teli\_honapok = {„Január”, „Február”, „December”}” (Ha helyileg megadjuk az adatot, akkor nem kell a „new” kulcsszó)
  + „double[] szamok = new double[] {3.14, 1.5, 2.5}” (ha helyileg meg van adva az adat, amit beviszünk inicializálásnál, akkor magától eldönti a tömb méretét)