

1. Egy szövegben háromféle zárójelpár szerepelhet:

- ()
- []
- { }

Írj programot, amely beolvasson egy szöveget, majd megadja, hogy a zárójeljezése helyes-e! (Pl. hibás zárójeljezés az alábbi: XX[YY{ZZ}AA].)

hibás:=hamis

Ciklus i=1-től cdb-ig

 Be: c

 Ha c='[' vagy c='(' vagy c='{ ' akkor Verembe(c)

 Ha c=']' vagy c=')' vagy c='}' akkor

 Ha nem üres? akkor Veremből(cs)

 Ha c=']' és cs≠ '[' vagy c=')' és cs≠ '('
 vagy c='}' és cs≠ '{ ' akkor hibás:=igaz

 különben hibás:=igaz

Ciklus vége

Ha nem üres? akkor hibás:=igaz

2. Egy verembe különböző méretű sorozatokat helyezünk el úgy, hogy legfelül lesz a sorozat elemszáma, alatta pedig a sorozat elemei. Készítsd el a SorozatVerembe és a SorozatVeremből eljárásokat úgy, hogy mindegyiknek két paramétere van: a sorozat elemszáma, valamint a sorozat elemeit tartalmazó tömb! A megvalósításhoz használd az elemi veremkezelő eljárásokat!

SorozatVerembe(sdb,t):

 Ciklus i=1-től sdb-ig

 Verembe(t[i])

 Ciklus vége

 Verembe(sdb)

Eljárás vége.

SorozatVeremből(sdb,t):

 Veremből(sdb)

 Ciklus i=sdb-től 1-ig -1-esével

 Veremből(t[i])

 Ciklus vége

Eljárás vége.

3. Dinamikusan láncolt listához írd meg a Beszúrutolsónak műveletet az ábrázolás felhasználásával!

Írj hozzá tesztelő programot is!

Beszúrutolsónak(e):

 Elsőre

 Ciklus amíg nem utolsó?

 Következőre

 Ciklus vége

 Beszúrmögé(e)

Eljárás vége.

4. Egy szekvenciális ábrázolású sorban léptessük előre eggyel az n. elemet!

Előrelép(n):

 honnan:=első+n

 Ha honnan>max akkor honnan:=honnan-max

 hova:=első+n-1

 Ha hova>max akkor hova:=hova-max

 csere(tömb(honnan), tömb(hova))

Eljárás vége.

1. Egy szövegben háromféle zárójelpár szerepelhet:

- ()
- []
- { }

Írj programot, amely beolvasson egy szöveget, majd megadja, hogy a zárójelezése helyes-e! (Pl. hibás zárójelezés az alábbi: XX[YY{ZZ}AA].)

hibás:=hamis

Ciklus i=1-től cdb-ig

 Be: c

 Ha c='[' vagy c='(' vagy c='{ ' akkor Verembe(c)

 Ha c=']' vagy c=')' vagy c='}' akkor

 Ha nem üres? akkor Verembol(cs)

 Ha c=']' és cs≠ '[' vagy c=')' és cs≠ '('
 vagy c='}' és cs≠ '{' akkor hibás:=igaz

 különben hibás:=igaz

Ciklus vége

Ha nem üres? akkor hibás:=igaz

2. Egy sorba különböző méretű számsorozatokot helyezünk el úgy, hogy először a sorozat elemei kerüljenek a sorba, amit egy 0 értékű elem zár le. Készítsd el a SorozatSorba és a SorozatSorból eljárásokat úgy, hogy mindegyiknek két paramétere van: a sorozat elemszáma, valamint a sorozat elemeket tartalmazó tömb! A megvalósításhoz használd az elemi sorkezelő eljárásokat!

SorozatSorba(sdb,t):

 Ciklus i=1-től sdb-ig

 Sorba(t[i])

 Ciklus vége

 Sorba(0)

Eljárás vége.

SorozatSorból(sdb,t):

 sdb:=0

 Sorból(s)

 Ciklus amíg s>0

 sdb:=sdb+1; t[sdb]:=s; Sorból(s)

 Ciklus vége

Eljárás vége.

3. Dinamikusan láncolt listához írd meg az Utolsóttöröl műveletet az ábrázolás felhasználásával! Írj hozzá tesztelő programot is!

Utolsóttöröl:

 Elsőre

 Ciklus amíg nem utolsó?

 Következőre

 Ciklus vége

 Kihagy

Eljárás vége.

4. Egy szekvenciális ábrázolású sorban léptessük hátra eggyel az n. elemet!

Hátralép(n):

 honnan:=első+n

 Ha honnan>max akkor honnan:=honnan-max

 hova:=első+n+1

 Ha hova>max akkor hova:=hova-max

 csere(tömb(honnan), tömb(hova))

Eljárás vége.