

1. kérdés

1 / 1 pont

A bináris fa melyik bejárásának felel meg az alábbi algoritmus?

bejárás (t:BinTree)	
t ≠ ∅	
bejárás (t→left)	
bejárás (t→right)	SKIP
process(t)	

Helyes!

☒ posztorder

☐ preorder

☐ inorder

2. kérdés

1 / 1 pont

Adja meg az alábbi fa inorder bejárását! A számokat vesszővel válassza el!

```
graph TD; 6((6)) --> 2((2)); 6 --> 7((7)); 2 --> 3((3)); 2 --> 4((4)); 3 --> 4; 4 --> 9((9)); 4 --> 2((2)); 7 --> 5((5)); 5 --> 1((1)); 1 --> 8((8));
```

Helyes!

3,9,4,2,2,6,7,1,8,5

Helyes válaszok

3,9,4,2,2,6,7,1,8,5

3. kérdés

1 / 1 pont

Az üres fa mélységét 0-nak definiáljuk.

☐ Igaz

☒ Hamis

4. kérdés

1 / 1 pont

Hogyan tesztljük, hogy egy csúcs levél a fában?

☐ t->left = 0 v t->right=0

☒ t->left = 0 ∧ t->right=0

☐ t->left ≠ 0 v t->right≠0

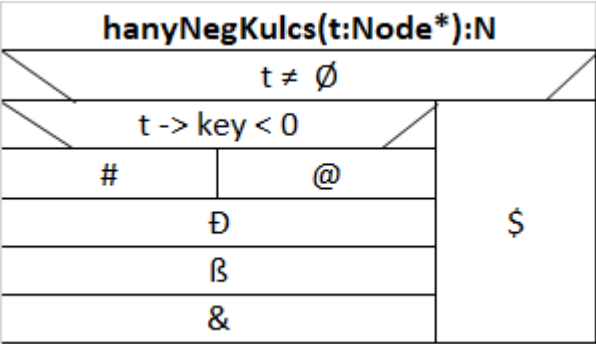
- ☐ $t \rightarrow \text{left} \neq 0 \wedge t \rightarrow \text{right} \neq 0$
- ☐ $(t \rightarrow \text{left} \neq 0 \wedge t \rightarrow \text{right} = 0) \vee (t \rightarrow \text{left} = 0 \wedge t \rightarrow \text{right} \neq 0)$

5. kérdés

5 / 5 pont

Párosítsa össze a struktogramban megjelölt helyeket a megfelelő utasítással!

A feladat, hogy számoljuk meg, hogy hány negatív kulcs van egy egész számokat tartalmazó bináris fában! A megoldás során preorder bejárást alkalmazunk!



Helyes!

#

c:=1

Helyes!

@

c:=0

Helyes!

Đ

c:=c+hanyNegKulcs(t->left)

Helyes!

ß

c:=c+hanyNegKulcs(t->right)

Helyes!

&

return c

Helyes!

\$

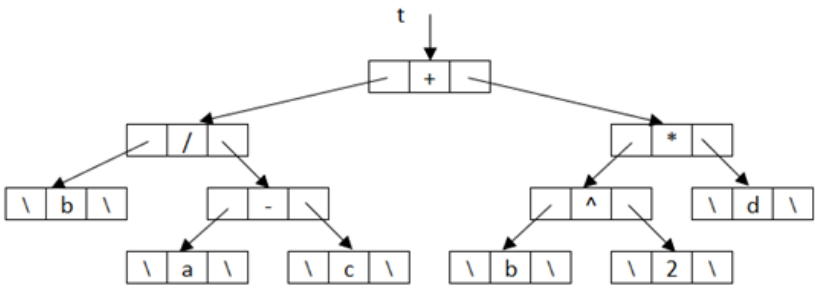
return 0

6. kérdés

1 / 1 pont

Adjuk meg az alábbi kifejezésfának megfelelő teljesen zárójelezett kifejezést!

(Ne tegyünk szóközőket a kifejezésbe, viszont minden összetett (rész)kifejezést zárójelezzünk bel!)



Helyes!

((b/(a-c))+((b^2)*d))

Helyes válaszok

((b/(a-c))+((b^2)*d))