

## Struktury Drzewiaste

Przyjmujemy następującą definicję drzewa binarnego.

---

C++

---

```
struct ele {  
    int dane;  
    ele * lewy;  
    ele * prawy;  
}
```

---

Zad. 1. Zapisz rekurencyjnie metody:

PREORDER, POSTORDER, INORDER

przeglądania drzewa binarnego.

Zad. 2. Oblicz liczbę węzłów w drzewie binarnym.

Zad. 3. Oblicz wysokość drzewa binarnego

int wysokość(ele\* korzen)

Zad. 4. Podaj adres elementu maksymalnego w drzewie binarnym.

Zad. 5. Z zapisanego poniżej ciągu liczb utwórz drzewo BST (drzewo poszukiwań binarnych).

2, 1, 30, 10, 5, 4, 11, 18, 7, 40

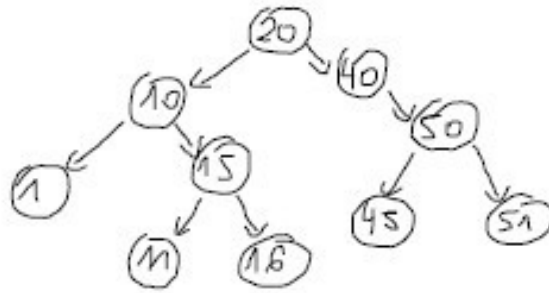
Nie pisz algorytmu tylko wynik przedstaw graficznie.

Zad. 6. Zaimplementuj wyszukiwanie elementu o zadanym kluczu w drzewie BST.

Zad. 7. Zaimplementuj dodawanie i usuwanie zadanego elementu ze struktury drzewa BST.

Zad. 8. Napisz funkcję czyszczącą drzewo binarne z pamięci.

Zad. 9. Z drzewa BST postaci:



usuń klucz 10, a następnie klucz 40.

Wynik przedstaw graficznie, czyli jak wygląda drzewo po 2 operacjach usunięcia klucza.

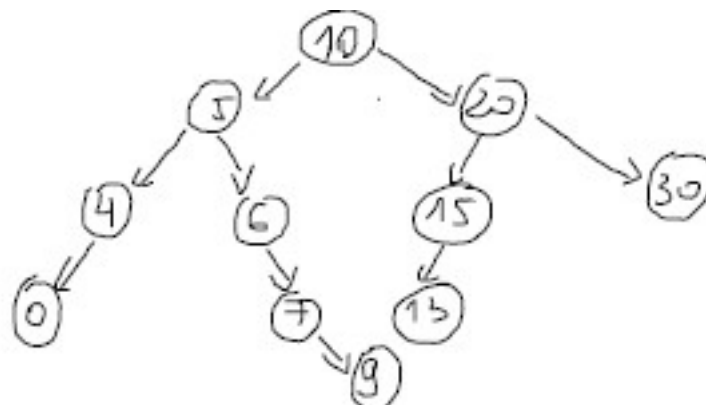
Zad. 10. Zapisz funkcję wypisującą dane drzewo binarne poziomami (ma wypisywać wartości kluczy znajdujące się w węzłach drzewa).

Zad. 11. Napisz funkcję sprawdzającym czy dany element występuje w drzewie binarnym.

Zad. 12. Napisz funkcję sprawdzającym czy dane drzewo binarne jest drzewem AVL (wyważonym). Drzewo jest wyważone wtedy tylko wtedy, gdy dla każdego węzła wysokości dwóch jego poddrzew różnią się od co najwyżej 1.

Zad. 13. Napisz funkcję sprawdzającym czy dane drzewo binarne jest drzewem BST.

Zad. 14. Zastosuj procedury PREORDER, POSTORDER, INORDER do poniższego drzewa:



Zad. 15. Zastosuj procedury PREORDER, POSTORDER, INORDER do drzewa z zadania 14 zakładając, że odwiedzenie wierzchołka p wiąże się z następującym działaniem:

```
If (p->lewy != 0 && (p->dane) - (p->lewy->dane)<2)
    p->lewy->dane += 2;
If (p->lewy == 0)
    p->prawy = 0;
```

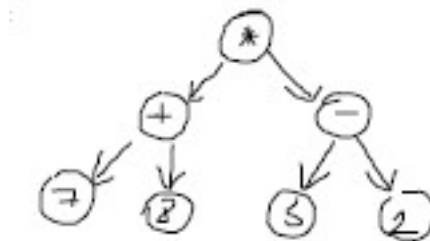
Zad. 16. Napisz funkcję, która tworzy kopie podanego drzewa binarnego.

Zad. 17. Strukturę drzewiastą zdefiniowano w następujący sposób:

```
struct node{  
    char op;  
    int num;  
    node* left;  
    node* right;  
}
```

Drzewa zdefiniowane w ten sposób mogą reprezentować wyrażenia arytmetyczne.

Np. wyrażenie  $(7+8)*(3-2)$  można przedstawić jako następujące drzewo:



Napisz funkcję, która oblicza wartości takich drzew.

Zad. 18. Przyjmijmy, że w BST znajdują się liczby od 1 do 1000 i chcemy wyszukać liczbę 363. Które z poniższych ciągów wierzchołków nie mogą zostać sprawdzone w procedurze SZUKAJ:

- a) 2, 252, 401, 398, 330, 344, 397, 363.
- b) 924, 220, 911, 244, 858, 258, 362, 363.
- c) 925, 202, 911, 240, 912, 245, 363.
- d) 2, 399, 387, 219, 266, 382, 381, 278, 363.
- e) 935, 278, 347, 621, 299, 392, 358, 363.

Zadanie 19. Opisz jak można obliczyć wysokość drzewa BST na podstawie kluczy, z których go zbudowano (bez budowania tego drzewa!).