

Zestaw 2B

Proszę skorzystać z podanej poniżej ramówki. W razie potrzeby proszę odpowiednio zmodyfikować listy argumentów funkcji i typy wartości zwracanych przez funkcje. Do każdego zadania należy przygotować funkcje umożliwiające wyświetlanie wyników (zgodnie z ramówką).

Zadanie 1:

4pkt

Zaimplementuj operacje push i pop dla stosu reprezentowanego w postaci tablicy. Stos przechowuje znaki. Implementacja optymalna.

Zadanie 2:

6pkt

Dla drzewa BST reprezentowanego w postaci dowiązań zaimplementuj:

- operację dodawania węzła, oraz
- operację wyświetlania wartości przechowywanych w węzłach w porządku inorder.

Drzewo przechowuje liczby. Implementacja optymalna.

Ramówka do zadań:

```
//Zadanie 1
void push(char z){}
char pop(){}
void print(){} //wydruk stanu stosu
void zadanie1(){
    cout<<"Zadanie 1"<<endl;
    push('a'); print(); push('e'); print(); push('r'); print(); push('n'); print(); push('e'); print();
    cout<<pop();print(); cout<<pop();print();
    push('r'); print(); push('x'); print();
    cout<<pop();print(); cout<<pop();print(); cout<<pop(); print();
    cout<<pop();print(); cout<<pop();print(); cout<<pop();print(); cout<<pop(); print();
    cout<<"Zadanie 1 - koniec"<<endl;
}

//Zadanie 2
void dodaj(int z){}
void inorder(){}
void zadanie2(){
    cout<<"Zadanie 2"<<endl;
    dodaj(10); dodaj(15); dodaj(6); dodaj(7); dodaj(3); dodaj(5); dodaj(12); inorder();
    cout<<"Zadanie 3 - koniec"<<endl;
}

int main(int argc, char *argv[])
{
    zadanie1();
    zadanie2();

    system("PAUSE");
    return EXIT_SUCCESS;
}
```