

Zestaw 2A

Ocena:

ocena 3.0	Ocena 4.0	Ocena 5.0
Zad. 1 lub Zad. 2 lub Zad. 3	Zad. 1, Zad. 2 lub Zad. 1, Zad. 3 lub Zad. 2, Zad. 3	Zad. 1, Zad. 2, Zad. 3

Ramówka programu na następnej stronie. Proszę z niej skorzystać. W razie potrzeby proszę odpowiednio zmodyfikować listy argumentów funkcji i typy wartości zwracanych przez funkcje.

Zadanie 1:

Zaimplementuj operacje push i pop dla stosu przechowującego znaki. Dowolna reprezentacja stosu. Należy umożliwić także wyświetlenie stanu stosu. Implementacja optymalna.

Zadanie 2:

Zaimplementuj operację wyświetlania wartości przechowywanych w drzewie BST w porządku inorder. Reprezentacja tablicowa. Znak '0' powinien reprezentować NULL. Implementacja optymalna.

Zadanie 3:

Zaimplementuj operację usuwania wszystkich elementów o podanej wartości klucza z jednokierunkowej listy przechowującej znaki. Należy także utworzyć listę (dowolny sposób) i umożliwić wyświetlenie jej stanu przed i po usunięciu elementu. Implementacja optymalna.

Ramówka programu 2A:

```
#include <cstdlib>
#include <iostream>

using namespace std;

//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 inorder(char bst[]){}
void zadanie2(){
    cout<<"Zadanie 2"<<endl;
    //drzewo BST użyte w zadaniu powinno być wynikiem dodawania kolejno następujących znaków: l d h p j a o
    char bst[] ={}; //należy w odpowiedni sposób zainicjalizować tablicę podanymi znakami
    inorder(bst) ;
    cout<<"Zadanie 2 - koniec"<<endl;
}

//Zadanie 3

void usun(char z){}
void utworzL(){}
void printL(){}
void zadanie3(){
    cout<<"Zadanie 3"<<endl;
    //przed usuwaniem lista powinna zawierać: a x r w x t a a
    printL(); usun('w'); printL(); usun('x'); printL(); usun('a'); printL();
    cout<<"Zadanie 3 - koniec"<<endl;
}

int main(int argc, char *argv[])
{
    zadanie1();
    zadanie2();
    zadanie3();

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