Zestaw 2D

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 liczby naturalne. Dowolna reprezentacja stosu. Należy umożliwić także wyświetlenie stanu stosu. Implementacja optymalna.

Zadanie 2:
Zaimplementuj operację wyszukiwania w drzewie BST, przechowującym liczby naturalne, następnika węzła o podanej wartości. Wynikiem na być wartość przechowywana w węźle lub -1 jeśli nie istnieje. Reprezentacja tablicowa. Implementacja optymalna.

Zadanie 3:
Zaimplementuj operację usuwania całej listy, dla listy jednokierunkowej. Lista przechowuje liczby naturalne. Należy także umożliwić wyświetlenie stanu listy przed i po usunięciu listy. Implementacja optymalna.

Ramówka program 2D:

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

using namespace std;

//Zadanie 1

void push(int z){}
int pop(){}
void print(){} //wydruk stanu stosu
void zadanie1(){
cout<<”Zadanie 1”<<endl;
push(1); print(); push(4); print(); push(9); print(); push(2); print(); push(5); print();
cout<<pop();print(); cout<<pop();print();
push(3); print(); push(7); 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

int nastepnik(char bst[], int s){}
void zadanie2(){
cout<<”Zadanie 2”<<endl;
//drzewo BST użyte w zadaniu powinno być wynikiem dodawania kolejno następujących liczb: 9 5 8 12 6 10 2
char bst[] ={};  //należy w odpowiedni sposób zainicjalizować tablicę podanymi znakami
cout<<nastepnik(bst,5);  cout<<nastepnik(bst,6); cout<<nastepnik(bst,12);
cout<<”Zadanie 2 - koniec”<<endl;
}

//Zadanie 3

void usun(){}
void printL(){}
void zadanie3(){
cout<<”Zadanie 3”<<endl;
printL(); usun(); printL();
cout<<”Zadanie 3 - koniec”<<endl;
}

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

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