№1) Создать стек, поместить в нее 7 целых чисел, вывести первый элемент стека, напечатать весь стек.

#include <iostream>

#include <queue>

#include <stack>

#include <string>

using namespace std;

int main() {

stack<int> st;

for (int i = 0; i < 7; i++) {

int val;

cin >> val;

st.push(val);

}

cout << "The first (upper) element: " << st.top() << endl;

while (!st.empty()) {

cout << st.top() << " ";

st.pop();

}

}

№2) Найти ошибки:

#include <iostream>

**#include <queue> // не подключена библиотека очереди**

using namespace std;

int main() {

setlocale(LC\_ALL,"rus");

queue <int> q;

cout << "Пользователь, пожалуйста, введите 7 чисел: " << endl;

for (int h = 0; h < 7; h++) {

**int** a; // **int вместо char, иначе в стек пойдут коды символов чисел вместо значений**

cin >> a;

q.push(a);

}

cout << endl;

cout << "Самый первый элемент в очереди: " << q.**front()** << endl; // нет метода first()

q.pop();

cout << "Новый первый элемент (после удаления): " << q.**front()** << endl; // нет first()

if (!q.empty()) cout << "Очередь не пуста!";

system("pause");

return 0;

}

№3) Поместить в очередь 10 целых чисел. После извлекать из очереди значение. Если полученный элемент – четное отрицательное число, поместить его в бинарное дерево.

· На вход поступила последовательность (1, -10, 0, 9, 11, -6, -2, 4, -1, -5). Нарисовать полученное в ходе задачи дерево

· Реализовать добавление элемента в дерево, печать дерева (порядок обхода симметричный), подсчет суммы элементов дерева

#include <iostream>

#include <string>

#include <queue>

using namespace std;

struct Node {

int x;

Node\* left, \* right;

};

typedef Node\* TreePtr;

void addNode(TreePtr& t, int value) {

if (t == NULL) {

t = new Node;

t->x = value;

t->left = NULL;

t->right = NULL;

}

else if (value < t->x) addNode(t->left, value);

else if (value > t->x) addNode(t->right, value);

}

void inOrder(TreePtr t, int lvl = -1) {

if (t) {

if (lvl == -1) {

inOrder(t->left);

cout << t->x << " ";

inOrder(t->right);

}

else {

inOrder(t->right, lvl + 1);

for (int i = 0; i < lvl; i++) cout << " ";

cout << t->x << endl;

inOrder(t->left, lvl + 1);

}

}

}

int sum(TreePtr tree) {

int res = 0;

TreePtr p = tree;

if (p->left) res += sum(p->left);

res += p->x;

if (p->right) res += sum(p->right);

return res;

}

int main()

{

TreePtr tree = NULL;

queue<int> qu;

const int n = 10;

int x;

for (int i = 0; i < n; i++) {

cout << "Enter " << i+1 << " number: ";

cin >> x;

qu.push(x);

cout << endl;

}

while (!qu.empty()) {

x = qu.front();

if ((x % 2 == 0) && (x < 0))

addNode(tree, x);

qu.pop();

}

cout << "tree:" << endl;

inOrder(tree);

cout << endl << endl;

inOrder(tree, 0); // наглядный вывод

cout << "sum: " << sum(tree) << endl;

}

1, -10, 0, 9, 11, -6, -2, 4, -1, -5 🡪 -10

-6

-2

№4) С помощью стека решить задачу перевода десятичного числа, записанного как строка символов в двоичный код.

#include <iostream>

#include <queue>

#include <stack>

#include <string>

using namespace std;

int main() {

string sequence;

cin >> sequence;

stack<char> st;

int dec = atoi(sequence.c\_str());

while (dec) {

st.push('0' + dec % 2);

dec /= 2;

}

string binresult;

while (!st.empty()) {

cout << st.top();

binresult += st.top();

st.pop();

}

cout << endl;

dec = atoi(sequence.c\_str());

cout << dec << endl;

char proverka[10];

\_itoa\_s<10>(dec, proverka, 2);

cout << proverka << endl;

}