**BÁO CÁO PROJECT 1**

**WEEK 3**

Nội dung

[**Hoàn Thành 8/8** 2](#_Toc181108505)

[**Problem: Week 3 - BST\_INSERTION\_MANIPULATION** 2](#_Toc181108506)

[**Problem: Week 3 - PARENTHESIS\_CHECK** 5](#_Toc181108507)

[**Problem: Week 2** FAMILY\_TREE 8](#_Toc181108508)

[**Problem: Week 3 - LINKED\_LIST\_MANIPULATION** 11](#_Toc181108509)

[**Problem: Week 3** QUEUE\_SIM 16](#_Toc181108510)

[**Problem: Week 3 - STACK\_SIM** 18](#_Toc181108511)

[**Problem: Week 3 - TREE\_MANIPULATION\_TRAVERSAL** 21](#_Toc181108512)

[**Problem: Week 3 - WATERJUG** 25](#_Toc181108513)

# **Hoàn Thành 8/8**

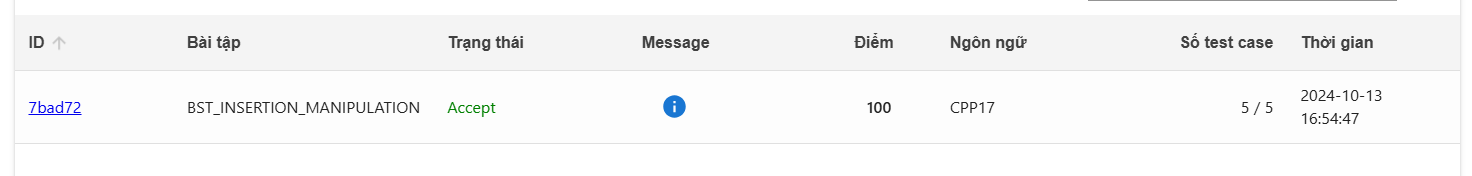
A screenshot of a computer

Description automatically generated

A screenshot of a phone

Description automatically generated

# **Problem: Week 3 - [BST\_INSERTION\_MANIPULATION](https://hustack.soict.ai/programming-contest/student-view-contest-problem-detail/PROJECT_1_2024_1/BST_INSERTION_MANIPULATION" \t "_blank)**



#include <bits/stdc++.h>

#define Task "bai1"

#define int long long

#define maxn 100005

using namespace std;

int dd[maxn], d[maxn][2],root = -1e18;

void add(int u, int v)

{

if (v < u)

{

if (d[u][0] == -1e18) d[u][0] = v;

else add(d[u][0], v);

}

else

{

if (d[u][1] == -1e18) d[u][1] = v;

else add(d[u][1], v);

}

}

void BST(int u)

{

cout << u << " ";

if (d[u][0] != -1e18) BST(d[u][0]);

if (d[u][1] != -1e18) BST(d[u][1]);

}

main()

{

ios::sync\_with\_stdio(false);

cin.tie(0);

cout.tie(0);

if (fopen(Task".inp", "r"))

{

freopen(Task".inp", "r", stdin);

freopen(Task".out", "w", stdout);

}

while (1)

{

string s;

cin >> s;

if (s == "insert")

{

int u;

cin >> u;

if (!dd[u])

{

dd[u] = 1;

d[u][0] = d[u][1] = -1e18;

if (root == -1e18) root = u;

else add(root, u);

}

}

else break;

}

BST(root);

}

# **Problem: Week 3 - [PARENTHESIS\_CHECK](https://hustack.soict.ai/programming-contest/student-view-contest-problem-detail/PROJECT_1_2024_1/PARENTHESIS_CHECK" \t "_blank)**

A screenshot of a computer

Description automatically generated

#include <bits/stdc++.h>

#define Task "bai1"

#define int long long

#define maxn 200005

using namespace std;

long long n, a, b;

string base = "()[]{}";

bool check(string s)

{

stack <int> st;

for (char c : s)

{

int x = base.find(c);

if (x % 2)

{

if (st.empty() ||st.top() != x - 1) return 0;

st.pop();

}

else st.push(x);

}

return st.empty();

}

main()

{

ios\_base::sync\_with\_stdio(0);

cout.tie(0);

cin.tie(0);

if (fopen(Task".inp", "r"))

{

freopen(Task".inp", "r", stdin);

freopen(Task".out", "w", stdout);

}

string s;

cin >> s;

{

if (check(s))

{

cout << 1;

return 0;

}

cout << 0;

}

}

# **Problem: Week 2** [FAMILY\_TREE](https://hustack.soict.ai/programming-contest/student-view-contest-problem-detail/PROJECT_1_2024_1/FAMILY_TREE)

A screenshot of a phone

Description automatically generated

#include <bits/stdc++.h>

#define Task "bai1"

#define int long long

#define maxn 100005

using namespace std;

map <string, int> Map;

int par[maxn], n, high[maxn], child[maxn];;

vector <int> adj[maxn];

void dfs(int u)

{

for(int i = 0; i < adj[u].size(); i++)

{

int v = adj[u][i];

dfs(v);

child[u] += (child[v] + 1);

high[u] = max(high[u], high[v] + 1);

}

}

main()

{

ios\_base::sync\_with\_stdio(0);

cout.tie(0);cin.tie(0);

if (fopen(Task".inp", "r"))

{

freopen(Task".inp", "r", stdin);

freopen(Task".out", "w", stdout);

}

while (1)

{

string s;

cin >> s;

if (s == "\*\*\*") break;

int u = Map[s];

if (!u) u = ++n;

Map[s] = u;

cin >> s;

int v = Map[s];

if (!v) v = ++n;

Map[s] = v;

par[u] = v;

adj[v].push\_back(u);

}

for (int i = 1; i <= n; i++) if (!par[i]) dfs(i);

while (1)

{

string s;

cin >> s;

if (s == "\*\*\*") break;

if (s == "descendants")

{

cin >> s;

cout << child[Map[s]] << "\n";

}

else

{

cin >> s;

cout << high[Map[s]] << "\n";

}

}

}

# **Problem: Week 3 - [LINKED\_LIST\_MANIPULATION](https://hustack.soict.ai/programming-contest/student-view-contest-problem-detail/PROJECT_1_2024_1/LINKED_LIST_MANIPULATION" \t "_blank)**

A screenshot of a computer

Description automatically generated

#include <bits/stdc++.h>

#define Task "bai1"

#define int long long

#define maxn 100005

using namespace std;

int n, a[maxn], trc[maxn], sau[maxn], check[maxn], cuoi = 1e5;

void add(int u, int v, int w)

{

sau[v] = u;

trc[w] = u;

trc[u] = v;

sau[u] = w;

}

void out()

{

int u = sau[0];

while (u != cuoi)

{

cout << u <<" ";

u = sau[u];

}

}

main()

{

ios::sync\_with\_stdio(false);

cin.tie(0);

cout.tie(0);

if (fopen(Task".inp", "r"))

{

freopen(Task".inp", "r", stdin);

freopen(Task".out", "w", stdout);

}

cin >> n;

for (int i = 1; i <= n; i++)

{

cin >> a[i];

check[a[i]] = 1;

}

a[n + 1] = cuoi;

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

{

sau[a[i]] = a[i + 1];

trc[a[i + 1]] = a[i];

}

while (1)

{

string s;

cin >> s;

if (s == "addlast")

{

int u;

cin >> u;

if (!check[u])

{

check[u] = 1;

add(u, trc[cuoi], cuoi);

}

}

else if (s == "addfirst")

{

int u;

cin >> u;

if (!check[u])

{

check[u] = 1;

add(u, 0, sau[0]);

}

}

else if (s == "addafter")

{

int u, v;

cin >> u >> v;

if (check[v] && !check[u])

{

check[u] = 1;

add(u, v, sau[v]);

}

}

else if (s == "addbefore")

{

int u, v;

cin >> u >> v;

if (check[v] && !check[u])

{

check[u] = 1;

add(u, trc[v], v);

}

}

else if (s == "remove")

{

int u;

cin >> u;

if (check[u])

{

check[u] = 0;

int v = trc[u], c = sau[u];

sau[v] = c;

trc[c] = v;

}

}

else if (s == "reverse")

{

int u = sau[0];

n = 0;

while (u != cuoi)

{

a[++n] = u;

u = sau[u];

}

for (int i = 1; i \* 2 <= n; i++) swap(a[i], a[n - i + 1]);

a[n + 1] = cuoi;

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

{

sau[a[i]] = a[i + 1];

trc[a[i + 1]] = a[i];

}

}

else

break;

}

out();

return 0;

}

# **Problem: Week 3** [QUEUE\_SIM](https://hustack.soict.ai/programming-contest/student-view-contest-problem-detail/PROJECT_1_2024_1/QUEUE_SIM)

A screenshot of a phone

Description automatically generated

#include <bits/stdc++.h>

#define Task "bai1"

#define int long long

#define maxn 1000005

using namespace std;

int a[505][505], sumcot[505];

queue<int> st;

main()

{

ios\_base::sync\_with\_stdio(0);

cout.tie(0);

cin.tie(0);

if (fopen(Task".inp", "r"))

{

freopen(Task".inp", "r", stdin);

freopen(Task".out", "w", stdout);

}

string s;

while(cin >> s)

{

int t;

if (s == "#") return 0;

if(s == "PUSH")

{

cin >> t;

st.push(t);

}

else

{

if(st.empty() == 1)

{

cout << "NULL" <<"\n";

continue;

}

t = st.front();

cout << t;

st.pop();

cout <<"\n";

}

}

}

# **Problem: Week 3 - [STACK\_SIM](https://hustack.soict.ai/programming-contest/student-view-contest-problem-detail/PROJECT_1_2024_1/STACK_SIM" \t "_blank)**

A screenshot of a computer

Description automatically generated

#include <bits/stdc++.h>

#define Task "bai1"

#define int long long

#define maxn 1000005

using namespace std;

int a[505][505], sumcot[505];

stack<int> st;

main()

{

ios\_base::sync\_with\_stdio(0);

cout.tie(0);

cin.tie(0);

if (fopen(Task".inp", "r"))

{

freopen(Task".inp", "r", stdin);

freopen(Task".out", "w", stdout);

}

string s;

while(cin >> s)

{

int t;

if (s == "#") return 0;

if(s == "PUSH")

{

cin >> t;

st.push(t);

}

else

{

if(st.empty() == 1)

{

cout << "NULL" <<"\n";

continue;

}

t = st.top();

cout << t;

st.pop();

cout <<"\n";

}

}

}

# **Problem: Week 3 - [TREE\_MANIPULATION\_TRAVERSAL](https://hustack.soict.ai/programming-contest/student-view-contest-problem-detail/PROJECT_1_2024_1/TREE_MANIPULATION_TRAVERSAL" \t "_blank)**

A screenshot of a computer

Description automatically generated

#include <bits/stdc++.h>

#define Task "bai1"

#define int long long

#define maxn 100005

using namespace std;

vector <int> adj[maxn];

int root, node[maxn], cha[maxn];

void PreOrder(int u, int p)

{

cout << u << " ";

for(int i = 0; i < adj[u].size(); i++)

{

int v = adj[u][i];

if (v != p) PreOrder(v, u);

}

}

void PostOrder(int u, int p)

{

for(int i = 0; i < adj[u].size(); i++)

{

int v = adj[u][i];

if (v != p) PostOrder(v, u);

}

cout << u <<" ";

}

void InOrder(int u, int p)

{

if (adj[u].size() < 2)

{

cout << u <<" ";

return;

}

bool check1 = 0;

for(int i = 0; i < adj[u].size(); i++)

{

int v = adj[u][i];

if (v != p)

{

InOrder(v, u);

if (!check1)

{

check1 = 1;

cout << u <<" ";

}

}

}

}

main()

{

ios::sync\_with\_stdio(false);

cin.tie(0);

cout.tie(0);

if (fopen(Task".inp", "r"))

{

freopen(Task".inp", "r", stdin);

freopen(Task".out", "w", stdout);

}

while (1)

{

string s;

cin >> s;

if (s == "MakeRoot")

{

int u;

cin >> u;

root = u;

node[u] = 1;

}

else if (s == "Insert")

{

int u, v;

cin >> u >> v;

if (node[v] && !node[u])

{

adj[u].push\_back(v);

adj[v].push\_back(u);

node[u] = 1;

}

}

else if (s == "PreOrder")

{

PreOrder(root, 0);

cout << "\n";

}

else if (s == "InOrder")

{

InOrder(root, 0);

cout << "\n";

}

else if (s == "PostOrder")

{

PostOrder(root, 0);

cout << "\n";

}

else

break;

}

}

# **Problem: Week 3 - [WATERJUG](https://hustack.soict.ai/programming-contest/student-view-contest-problem-detail/PROJECT_1_2024_1/WATERJUG" \t "_blank)**

A screenshot of a phone

Description automatically generated

#include <bits/stdc++.h>

#define Task "bai1"

#define int long long

using namespace std;

int a[5], d;

struct Node

{

int F, S, dem;

};

bool tr[901][901];

main()

{

ios::sync\_with\_stdio(false);

cin.tie(0);

cout.tie(0);

if (fopen(Task".inp", "r"))

{

freopen(Task".inp", "r", stdin);

freopen(Task".out", "w", stdout);

}

cin >> a[1] >> a[2] >> d;

if(d > max(a[1], a[2]) || d % \_\_gcd(a[1], a[2]) != 0)

{

cout << -1;

return 0;

}

queue<Node> q;

q.push(Node{0, 0, 0});

tr[0][0] = 1;

while(!q.empty())

{

Node check = q.front();

q.pop();

if(check.F == d || check.S == d)

{

cout << check.dem;

return 0;

}

if(!tr[a[1]][check.S])

{

q.push(Node{a[1], check.S, check.dem +1});

tr[a[1]][check.S] = 1;

}

if(!tr[check.F][a[2]])

{

q.push(Node{check.F, a[2], check.dem +1});

tr[check.F][a[2]] = 1;

}

if(!tr[0][check.S])

{

q.push(Node{0, check.S, check.dem +1});

tr[0][check.S] = 1;

}

if(!tr[check.F][0])

{

q.push(Node{check.F, 0, check.dem +1});

tr[check.F][0] = 1;

}

int k = min(check.F, a[2] - check.S);

int newF = check.F - k;

int newS = check.S + k;

if(!tr[newF][newS])

{

q.push(Node{newF, newS, check.dem +1});

tr[newF][newS] = 1;

}

k = min(check.S, a[1] - check.F);

newF = check.F + k;

newS = check.S - k;

if(!tr[newF][newS])

{

q.push(Node{newF, newS, check.dem +1});

tr[newF][newS] = 1;

}

}

cout << -1;

}