NAME: SALAR WASIL

REG NO: SP23-BSE-041

SECTION: BSE3

#include *<bits/stdc++.h>*

#include *<iostream>*

#include*<stack>*

using namespace std;

struct Node {

    char player\_name[100];

    int jersey\_number;

    int goal;

    int got\_red\_card;

    int got\_yellow\_card;

    float pass\_accuracy;

    Node\* next;

};

stack<Node\*> team\_1;

stack<Node\*> team\_2;

stack<Node\*> team\_3;

stack<Node\*> team\_4;

stack<Node\*> team\_5;

*/// Info Add Function.*

void player\_info\_add() {

    int team;

    cout << *"Team Number: "*;

    cin >> team;

    cout << endl;

    Node\* new\_node = new Node();

    cout << *"Enter Player Name: "*;

    cin>>new\_node->player\_name;

    cout << endl;

    cout << *"Enter Jersey Number: "*;

    cin >> new\_node->jersey\_number;

    cout << endl;

    cout << *"Enter Goal Figure: "*;

    cin >> new\_node->goal;

    cout << endl;

    cout << *"Enter Red Card Figure: "*;

    cin >> new\_node->got\_red\_card;

    cout << endl;

    cout << *"Enter Yellow Card Figure: "*;

    cin >> new\_node->got\_yellow\_card;

    cout << endl;

    cout << *"Enter The Percentage of Pass Accuracy: "*;

    cin >> new\_node->pass\_accuracy;

    cout << endl;

    new\_node->next = nullptr;

    switch (team) {

        case 1:

            team\_1.push(new\_node);

            break;

        case 2:

            team\_2.push(new\_node);

            break;

        case 3:

            team\_3.push(new\_node);

            break;

        case 4:

            team\_4.push(new\_node);

            break;

        case 5:

            team\_5.push(new\_node);

            break;

        default:

            cout << *"This memory only takes 5 team data."*;

    }

}

*/// Display Function For Specific Match.*

void display\_team() {

    int team;

    cout << *"Which teams info you need to see: "*;

    cin >> team;

    cout << endl;

    stack<Node\*> temp\_stack;

    switch (team) {

        case 1:

            temp\_stack = team\_1;

            break;

        case 2:

            temp\_stack = team\_2;

            break;

        case 3:

            temp\_stack = team\_3;

            break;

        case 4:

            temp\_stack = team\_4;

            break;

        case 5:

            temp\_stack = team\_5;

            break;

        default:

            cout << *"Sorry, there is nothing to print.\n"*;

            return;

    }

    if (temp\_stack.empty()) {

        cout << *"There is no info in the Match-"* << team << *" list.\n"*;

        return;

    }

    cout << *"======================== Player info for team-"* << team << *" list ======================== "* << endl;

    cout << endl;

    while (!temp\_stack.empty()) {

        Node\* temp = temp\_stack.top();

        temp\_stack.pop();

        cout << *"Player Name: "* << temp->player\_name << endl;

        cout << *"Jersey Number: "* << temp->jersey\_number << endl;

        cout << *"Goal Figure: "* << temp->goal << endl;

        cout << *"Red Card Figure: "* << temp->got\_red\_card << endl;

        cout << *"Yellow Card Figure: "* << temp->got\_yellow\_card << endl;

        cout << *"Pass accuracy: "* << temp->pass\_accuracy << endl;

        cout << *"\n"*;

    }

    cout << *"\n"*;

}

*/// Search Function Using Player Name and Player Jersey Number.*

void search\_player() {

    char name[100];

    int number;

    cout << *"Enter Name: "*;

    cin>>name;

    cout << endl;

    cout << *"Enter Jersey Number: "*;

    cin >> number;

    cout << endl;

    stack<Node\*> temp\_stack[5] = {team\_1, team\_2, team\_3, team\_4, team\_5};

    bool found = false;

    for (int i = 0; i < 5; i++) {

        while (!temp\_stack[i].empty()) {

            Node\* temp = temp\_stack[i].top();

            temp\_stack[i].pop();

            if (strcmp(temp->player\_name, name) == 0 && temp->jersey\_number == number) {

                cout << endl;

                cout << *">>>>>>>>>>>>>> Team-"* << i + 1 << *" <<<<<<<<<<<<<<<"* << endl;

                cout << endl;

                cout << *"Player Name: "* << temp->player\_name << endl;

                cout << *"Jersey Number: "* << temp->jersey\_number << endl;

                cout << *"Goal Figure: "* << temp->goal << endl;

                cout << *"Red Card Figure: "* << temp->got\_red\_card << endl;

                cout << *"Yellow Card Figure: "* << temp->got\_yellow\_card << endl;

                cout << *"Pass accuracy: "* << temp->pass\_accuracy << *"%"* << endl;

                cout << endl;

                found = true;

                break;

            }

        }

        if (found) {

            break;

        }

    }

    if (!found) {

        cout << *"No Info available about the player.\n"*;

    }

}

*/// This is for Updating Player Info.*

void edit() {

    char name[100];

    int number, team;

    cout << *"Enter Player Name: "*;

    cin>>name;

    cout << endl;

    cout << *"Enter Jersey Number: "*;

    cin >> number;

    cout << endl;

    cout << *"Enter Team Number: "*;

    cin >> team;

    cout << endl;

    stack<Node\*> temp\_stack;

    switch (team) {

        case 1:

            temp\_stack = team\_1;

            break;

        case 2:

            temp\_stack = team\_2;

            break;

        case 3:

            temp\_stack = team\_3;

            break;

        case 4:

            temp\_stack = team\_4;

            break;

        case 5:

            temp\_stack = team\_5;

            break;

        default:

            cout << *"There is nothing to Edit.\n"*;

            return;

    }

    if (temp\_stack.empty()) {

        cout << *"There is nothing to Edit.\n"*;

        return;

    }

    bool found = false;

    stack<Node\*> temp\_buffer;

    while (!temp\_stack.empty()) {

        Node\* temp = temp\_stack.top();

        temp\_stack.pop();

        if (strcmp(temp->player\_name, name) == 0 && temp->jersey\_number == number) {

            found = true;

            cout << *"Enter Update Player Name: "*;

            cin>>temp->player\_name;

            cout << endl;

            cout << *"Enter Update Jersey Number: "*;

            cin >> temp->jersey\_number;

            cout << endl;

            cout << *"Enter Update Goal Figure: "*;

            cin >> temp->goal;

            cout << endl;

            cout << *"Enter Update Red Card Figure: "*;

            cin >> temp->got\_red\_card;

            cout << endl;

            cout << *"Enter Update Yellow Card Figure: "*;

            cin >> temp->got\_yellow\_card;

            cout << endl;

            cout << *"Enter The Update Percentage of Pass Accuracy: "*;

            cin >> temp->pass\_accuracy;

            cout << endl;

        }

        temp\_buffer.push(temp);

    }

*// Restore stack*

    while (!temp\_buffer.empty()) {

        temp\_stack.push(temp\_buffer.top());

        temp\_buffer.pop();

    }

    switch (team) {

        case 1:

            team\_1 = temp\_stack;

            break;

        case 2:

            team\_2 = temp\_stack;

            break;

        case 3:

            team\_3 = temp\_stack;

            break;

        case 4:

            team\_4 = temp\_stack;

            break;

        case 5:

            team\_5 = temp\_stack;

            break;

    }

    if (!found) {

        cout << *"Player not found.\n"*;

    }

}

*/// Delete Function is for removing a player info.*

void delete\_player() {

    char name[100];

    int team\_number, jersey\_number;

    cout << *"Enter Team Number: "*;

    cin >> team\_number;

    cout << endl;

    cout << *"Enter Player Name: "*;

    cin>> name;

    cout << endl;

    cout << *"Enter Jersey Number: "*;

    cin >> jersey\_number;

    cout << endl;

    stack<Node\*> temp\_stack;

    switch (team\_number) {

        case 1:

            temp\_stack = team\_1;

            break;

        case 2:

            temp\_stack = team\_2;

            break;

        case 3:

            temp\_stack = team\_3;

            break;

        case 4:

            temp\_stack = team\_4;

            break;

        case 5:

            temp\_stack = team\_5;

            break;

        default:

            cout << *"Invalid team number.\n"*;

            return;

    }

    if (temp\_stack.empty()) {

        cout << *"No players in teams "* << team\_number << endl;

        return;

    }

    stack<Node\*> temp\_buffer;

    bool found = false;

    while (!temp\_stack.empty()) {

        Node\* temp = temp\_stack.top();

        temp\_stack.pop();

        if (strcmp(temp->player\_name, name) == 0 && temp->jersey\_number == jersey\_number) {

            delete temp;

            found = true;

        } else {

            temp\_buffer.push(temp);

        }

    }

*// Restore stack*

    while (!temp\_buffer.empty()) {

        temp\_stack.push(temp\_buffer.top());

        temp\_buffer.pop();

    }

    switch (team\_number) {

        case 1:

            team\_1 = temp\_stack;

            break;

        case 2:

            team\_2 = temp\_stack;

            break;

        case 3:

            team\_3 = temp\_stack;

            break;

        case 4:

            team\_4 = temp\_stack;

            break;

        case 5:

            team\_5 = temp\_stack;

            break;

    }

    if (found) {

        cout << *"Player removed successfully.\n"*;

    } else {

        cout << *"Player not found.\n"*;

    }

}

void undo\_last\_addition() {

    int team;

    cout << *"Team Number to undo last addition: "*;

    cin >> team;

    cout << endl;

    Node\* removed\_player = nullptr;

    switch (team) {

        case 1:

            if (!team\_1.empty()) {

                removed\_player = team\_1.top();

                team\_1.pop();

            }

            break;

        case 2:

            if (!team\_2.empty()) {

                removed\_player = team\_2.top();

                team\_2.pop();

            }

            break;

        case 3:

            if (!team\_3.empty()) {

                removed\_player = team\_3.top();

                team\_3.pop();

            }

            break;

        case 4:

            if (!team\_4.empty()) {

                removed\_player = team\_4.top();

                team\_4.pop();

            }

            break;

        case 5:

            if (!team\_5.empty()) {

                removed\_player = team\_5.top();

                team\_5.pop();

            }

            break;

        default:

            cout << *"Invalid team number.\n"*;

            return;

    }

    if (removed\_player != nullptr) {

        cout << *"Removed player: "* << removed\_player->player\_name << *" (Jersey #"* << removed\_player->jersey\_number << *")\n"*;

        delete removed\_player;

    } else {

        cout << *"No player to remove from Team-"* << team << *".\n"*;

    }

}

void show() {

    stack<Node\*> teams[5] = {team\_1, team\_2, team\_3, team\_4, team\_5};

    for (int i = 0; i < 5; i++) {

        if (teams[i].empty()) {

            cout << *"There is No Info in the Team-"* << (i + 1) << *" list."* << endl;

            cout << endl;

        } else {

            cout << *"======================== Player Info list for Team-"* << (i + 1) << *" ======================== "* << endl;

            cout << endl;

            stack<Node\*> temp\_stack = teams[i];

            while (!temp\_stack.empty()) {

                Node\* temp = temp\_stack.top();

                temp\_stack.pop();

                cout << *"Player Name: "* << temp->player\_name << endl;

                cout << *"Jersey Number: "* << temp->jersey\_number << endl;

                cout << *"Goal Figure: "* << temp->goal << endl;

                cout << *"Red Card Figure: "* << temp->got\_red\_card << endl;

                cout << *"Yellow Card Figure: "* << temp->got\_yellow\_card << endl;

                cout << *"Pass accuracy: "* << temp->pass\_accuracy << *"%"* << endl;

                cout << *"\n"*;

            }

            cout << *"\n"*;

        }

    }

}

int main() {

    cout << *" ################    #########         ########             ########      ################          "* << endl;

    cout << *" ################    ##########        ##########         ##########      ################          "* << endl;

    cout << *" ###                 ###########       ###########       ###########      ####                      "* << endl;

    cout << *" ###                 ###########       ####    ####     ####    ####      ####                      "* << endl;

    cout << *" ########            #########         ####     ####   ####     ####      ################          "* << endl;

    cout << *" ########            #######           ####      #########      ####      ################          "* << endl;

    cout << *" ###                 ###               ####       #######       ####                  ####          "* << endl;

    cout << *" ###                 ###               ####        #####        ####                  ####          "* << endl;

    cout << *" ###                 ###               ####         ###         ####      ################          "* << endl;

    cout << *" ###                 ###               ####         ###         ####      ################          "* << endl;

    cout << *"####################################################################################################"* << endl;

    cout << *"####################################################################################################"* << endl;

    cout << endl;

    cout << *"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Welcome To The FootBall Player Management System \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"*;

    cout << endl;

    while (true) {

        int choice;

        cout << *"1. Add Player.\n"*;

        cout << *"2. Edit PLayer.\n"*;

        cout << *"3. Show PLayers.\n"*;

        cout << *"4. Display Specific Team Info.\n"*;

        cout << *"5. Search Player.\n"*;

        cout << *"6. Delete Player.\n"*;

        cout << *"7. Undo Last Addition.\n"*;

        cout << *"8. Exit.\n"*;

        cout << endl;

        cout << *"Enter Your Choice: "*;

        cin >> choice;

        cout << endl;

        switch (choice) {

            case 1:

                player\_info\_add();

                break;

            case 2:

                edit();

                break;

            case 3:

                show();

                break;

            case 4:

                display\_team();

                break;

            case 5:

                search\_player();

                break;

            case 6:

                delete\_player();

                break;

            case 7:

                undo\_last\_addition();

                break;

            case 8:

                cout << *" #######      ####      ####    #######       ####       ####      ##    "* << endl;

                cout << *" ########      ####    ####     ########       ####     ####       ##    "* << endl;

                cout << *" #########      ####  ####      #########       ####   ####        ##    "* << endl;

                cout << *" ########        #######        ########          #######          ##    "* << endl;

                cout << *" #####            ####          #####              #####           ##    "* << endl;

                cout << *" #######           ##           #######             ##             ##    "* << endl;

                cout << *" #########         ##           #########           ##                   "* << endl;

                cout << *" #########         ##           #########           ##             ##    "* << endl;

                cout << *" #######           ##           #######             ##             ##    "* << endl;

                cout << *" ########################################################################"* << endl;

                cout << *" ########################################################################"* << endl;

                cout << endl;

                printf( *" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Exited FPMS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"*);

                exit(0);

            default:

                cout << *"Invalid choice, please try again.\n"*;

        }

    }

    return 0;

}