

```

#include "Header.h"

int main()
{
    room player_input;
    player turn;
    bool player, cont, winner;

start:

    if (start_page())
        input_page(player_input);
    player = check_starter(player_input);
    who_starts(player);

    while (!check_finish(player_input))
    {
        if (player == true)
        {
            turn = main_page1(player_input, player);
            player_func(player_input, turn, &cont);
            player = main_page2(player_input, player, turn);
            winner = true;
        }
        else
        {
            player_input = com_func(player_input, &cont);
            player = main_page3(player_input, player);
            winner = false;
        }
    }
    if (winner_page(winner))
    {
        reset_all(player_input);
        goto start;
    }
    return 0;
}

```

```

#ifndef _Header_H_included_
#define _Header_H_included_

#include <iostream>
#include <cstdlib>
#include <vector>
#include <cmath>
#include <iomanip>
#include <algorithm>
#include <ctime>
#include <string>
using namespace std;

struct player
{
    int pile, quantity;
};
struct room
{
    int first, second, third, pile, quantity;
    int player_moves = 0, computer_moves = 0;
};

int max_size(const room & player_input)
{
    int max = player_input.first;
    if (player_input.second > max)
        max = player_input.second;
    if (player_input.third > max)
        max = player_input.third;

    vector<int> v;
    int r;
    while (max > 0)
    {
        r = max % 2;
        v.push_back(r);
        max = max / 2;
    }
    return v.size();
}

int pow(int x, int y) {
    int time = 1;
    if (y == 0)
        return 1;
    else {
        int i = 1;
        while (i <= y) {
            time *= x;
            i++;
        }
        x = time;
    }
    return x;
}

vector<int> conv(int a, vector<int> & v, int size)
{
    vector<int> c;

```

```

    int r;
    while (a>0)
    {
        r = a % 2;
        v.push_back(r);
        a = a / 2;
    }

    int count = size - v.size();
    while (count > 0)
    {
        v.push_back(0);
        count--;
    }

    int k = v.size() - 1;
    for (int a = k; a >= 0; a--)
    {
        c.push_back(v[a]);
    }
    v = c;
    return v;
}
int deconv(vector<int> &a)
{
    int sum = 0;
    int count = a.size() - 1;
    int i = 0;
    while (count >= 0)
    {
        sum += (a[i] * pow(2, count));
        count--;
        i++;
    }
    return sum;
}
room reset_all(room & player)
{
    room room;
    player = room;
    return player;
}
int stoint(const string & str)
{
    int size = str.size(), sum = 0;
    for (int i = 0; i < size; i++)
    {
        if ((int)str[i]>48 && (int)str[i] < 58)
            sum += ((int)str[i] - 48)*pow(10, size - i - 1);
        else break;
    }
    return sum;
}

bool start_page()
{
    system("cls");
}

```



```

        cout << "
" << endl;
        cout << "
" << endl;
        cout << "
" << endl;
        cout << "
" << endl;
        cout << "
" << endl;
        cout << "XXX      XXX  XXXXXX  XXX  XXX  XXX      XXXXX
XXXXX  XXXXXXX" << endl;
        cout << " XXX  XXX  XXXXXXXX  XXX  XXX  XXX  XXXXXXXX
XXXXXXXX  XXXXXXX" << endl;
        cout << " XXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX
XXX  XXX  " << endl;
        cout << " XXXXXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX
XXX  " << endl;
        cout << " XXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX
XXXXXXXX " << endl;
        cout << " XXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX
XXX  XXXXXX " << endl;
        cout << " XXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX
XXX  XXX  " << endl;
        cout << " XXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX
XXX  XXX  " << endl;
        cout << " XXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX  XXX
XXXX  XXXXXXX" << endl;
        cout << " XXX  XXXXXX  XXXXX  XXXXXX  XXXXX  XXXX
XXXXXXXX  XXXXXXX" << endl;
        cout << "
" << endl;
        cout << "
" << endl;
        cout << " DO YOU WANT TO START THIS GAME NOW?
" << endl;
        cout << " PRESS (1 + ENTER) TO RESTART THE GAME
" << endl;
        cout << " PRESS ( 0 + ENTER) EXIT
" << endl;
        cout << "
" << endl;
        cout << "
" << endl;
        cout << "
" << endl;
        cout << "
" << endl;
    }
    cin >> x;
    if (stoi(x) == 1)
        return true;
    else return false;
}
bool check_starter(const room &player_input)
{
    int size = max_size(player_input);
    cout << "maxsize=" << size << endl;
    vector<int> first, second, third, number;
    conv(player_input.first, first, size);

```



```

    cout << "      | ... | + | ... | + | ... | +
" << endl;
    cout << "      |      |+ |      |+ |      |+
" << endl;
    cout << "      +-----+      +-----+      +-----+
" << endl;
    cout << "
" << endl;
    cout << "          input the number of packs of matches you want to play !
" << endl;
    cout << "          the amount of packs in the first second and third piles:
" << endl;
    cin >> first >> second >> third;
    player_input.first = stoint(first);
    player_input.second = stoint(second);
    player_input.third = stoint(third);

    if (player_input.first >= 0 && player_input.second >= 0 && player_input.third >=
0)
    {
        cout << "computer manages itself who starts the game !" << endl;
    }
    else
    {
        while (stop == false)
        {
            cout << "
" << endl;
            cout << "          PLAYER MOVES:...
" << endl;
            cout << "          COMPUTER MOVES:...
" << endl;
            cout << "
            cout << "          +-----+      +-----+      +-
-----+
" << endl;
            cout << "          +      +|      +      +|      +
+|      " << endl;
            cout << "          +      + |      +      + |      +
+ |      " << endl;
            cout << "          +      + |      +      + |      +
+ |      " << endl;
            cout << "          +      + + +      + + +
+ +      " << endl;
            cout << "          +      + + +      + + +
+ +      " << endl;
            cout << "          +      + + +      + + +
+ +      " << endl;
            cout << "          +-----+      +      +-----+      +      +-----
--+ +
" << endl;
            cout << "          |      | + |      | + |
| +      " << endl;
            cout << "          | ... | + | ... | + | ...
| +      " << endl;
            cout << "          |      |+ |      |+ |
|+      " << endl;

```



```

        if (x[0])
            return true;
        return false;
    }

player main_page1(const room &room, bool &play)
{
    string pile, quantity;
    system("color 1e");
    player player;

start:
    system("cls");
    if (play)
    {
        cout << "
" << endl;
        cout << "                                PLAYER MOVES:" << setfill('.')
<< setw(3) << room.player_moves << endl;
        cout << "                                COMPUTER MOVES:" << setfill('.')
<< setw(3) << room.computer_moves << endl;
        cout << "
" << endl;
        cout << "
        +-----+           +-----+           +-----+
---+      " << endl;
        cout << "          +           +|           +           +|           +
+|      " << endl;
        cout << "          +           + |           +           + |           +
+ |      " << endl;
        cout << "          +           + |           +           + |           +
+ |      " << endl;
        cout << "          +           + + +           + + +           +
+      " << endl;
        cout << "          +           + + +           + + +           +
+      " << endl;
        cout << "          +           + + +           + + +           +
+      " << endl;
        cout << "          +           + + +           + + +           +
+      " << endl;
        cout << "          +-----+ + +-----+ + +-----+ +
" << endl;
        cout << "          |           | +           |           | +           | +
" << endl;
        cout << "          |           " << setfill('.') << setw(3) << room.first << "          | +
|           " << setfill('.') << setw(3) << room.second << "          | +           |           " << setfill('.')
<< setw(3) << room.third << "          | + " << endl;
        cout << "          |           |+           |           |+           |           |+
" << endl;
        cout << "          +-----+           +-----+           +-----+
" << endl;
        cout << "
" << endl;
        cout << "
        enter the number of packs and than the number of pile
" << endl;
        cout << "
        you have to take at least one pack of match from one
pile      " << endl;
        cout << "
        or you can take all the packs from one pile !
" << endl;
    }
}

```



```

    cout << "          +-----+ + +-----+ + +-----+ +
" << endl;
    cout << "          |          | + |          | + |          | +
" << endl;
    cout << "          |          " << setfill('.') << setw(3) << room.first << "          | +
" << setfill('.') << setw(3) << room.second << "          | +          |          " << setfill('.') <<
setw(3) << room.third << "          | + " << endl;
    cout << "          |          | +          | +          |          | +
" << endl;
    cout << "          +-----+          +-----+          +-----+
" << endl;
    cout << "
" << endl;
    cout << "          OK ! you took " << player.quantity << "packs of matches
from " << player.pile << "th pile          " << endl;
    cout << "          then it's computer's turn press any key + enter to continue
" << endl;
    cout << "
" << endl;

```

```

    char x;
    cin >> x;
    if (x)
        play = false;
    return play;
}

```

```

bool main_page3(const room &room, bool &play)
{

```

```

    system("cls");
    system("color 1e");
    if (play == false)
    {
        cout << "
" << endl;
        cout << "          PLAYER MOVES:" << setfill('.')
<< setw(3) << room.player_moves << endl;
        cout << "          COMPUTER MOVES:" << setfill('.')
<< setw(3) << room.computer_moves << endl;
        cout << "
" << endl;
        cout << "          +-----+          +-----+          +-----+
---+          " << endl;
        cout << "          +          +|          +          +|          +
+|          " << endl;
        cout << "          +          + |          +          + |          +
+ |          " << endl;
        cout << "          +          + |          +          + |          +
+ |          " << endl;
        cout << "          +          + + +          + + +          +
+          " << endl;
        cout << "          +          + + +          + + +          +
+          " << endl;
        cout << "          +          + + +          + + +          +
+          " << endl;
        cout << "          +          + + +          + + +          +
+          " << endl;

```

```

    cout << "          +-----+ + +-----+ + +-----+ +
" << endl;
    cout << "          |           | + |           | + |           | +
" << endl;
    cout << "          |           " << setfill('.') << setw(3) << room.first << "          | +
|           " << setfill('.') << setw(3) << room.second << "          | +           |           " << setfill('.')
<< setw(3) << room.third << "          | + " << endl;
    cout << "          |           | +           | +           |           | +
" << endl;
    cout << "          +-----+           +-----+           +-----+
" << endl;
    cout << "
" << endl;
    cout << "          COMPUTER took " << room.quantity << " packs of matches
from " << room.pile << "th pile " << endl;
    cout << "          then it's your turn ! press any key + enter to continue
" << endl;
    cout << "
" << endl;
    char x;
    cin >> x;
    if (x)
        play = true;
}
return play;
}

int find_index(room &room, vector<int> first, vector<int> second, vector<int> third,
int max_index)
{
    vector<int> vec;
    int max = 0;
    int index = 0;
    if (first[max_index] > 0)
        vec.push_back(deconv(first));
    if (second[max_index] > 0)
        vec.push_back(deconv(second));
    if (third[max_index] > 0)
        vec.push_back(deconv(third));
    for (int i = 0; i < vec.size(); i++)
    {
        if (vec[i]>max)
            max = vec[i];
    }

    if (room.first == max)
        index = 1;
    if (room.second == max)
        index = 2;
    if (room.third == max)
        index = 3;
    return index;
}

room com_func(room &room, bool * cont)
{
    bool stop = true;
    vector<int> first, second, third, number;
    int size = max_size(room);

```

```

conv(room.first, first, size);
conv(room.second, second, size);
conv(room.third, third, size);

int sum1 = 0, index = 0, max_index = 0;
while (index < size)
{
    sum1 = (first[index] + second[index] + third[index]);
    number.push_back(sum1 % 2);
    index++;
}

index = 0;
while (index < size)
{
    if (number[index] > 0)
    {
        max_index = index;
        break;
    }
    index++;
}
int x = find_index(room, first, second, third, max_index);
int i = 0;
switch (x)
{
case 1:
{
    while (i < size)
    {
        number[i] = (first[i] + number[i]) % 2;
        i++;
    }
    room.pile = 1;
    room.quantity = (deconv(first) - deconv(number));
    room.first -= room.quantity;
    break;
}
case 2:
{
    while (i < size)
    {
        number[i] = (second[i] + number[i]) % 2;
        i++;
    }
    room.pile = 2;
    room.quantity = (deconv(second) - deconv(number));
    room.second -= room.quantity;
    break;
}
case 3:
{
    while (i < size)
    {
        number[i] = (third[i] + number[i]) % 2;
        i++;
    }
}
}

```

```

        room.pile = 3;
        room.quantity = (deconv(third) - deconv(number));
        room.third -= room.quantity;
        break;
    }
}
if (room.first == 0 && room.second == 0 && room.third == 0)
    *cont = false;
room.computer_moves++;
return room;
}
room player_func(room &room, const player & player, bool * cont)
{
start:
    if (player.pile > 0 && player.pile < 4)
    {
        switch (player.pile)
        {
            case 1:
                if (player.quantity <= room.first)
                    room.first = room.first - player.quantity;
                break;
            case 2:
                if (player.quantity <= room.second)
                    room.second = room.second - player.quantity;
                break;
            case 3:
                if (player.quantity <= room.third)
                    room.third = room.third - player.quantity;
                break;
            default: break;
        }
    }
    else goto start;
    if (room.first == 0 && room.second == 0 && room.third == 0)
        *cont = false;
    room.player_moves++;
    return room;
}

#endif _Header_H_included_#pragma once

```


