**Name : Shihu**

**Reg No. : 2020CA089**

#include <iostream> #include <map> #include <random>

std::default random engine generator; std::uniform int distribution<int> dice(1, 6); int rollDice() (

return dice(generator);

const bool sixesThrowAgain = true; const std::map<int, int> snl(

(4, 14},

(9, 31},

(17, 7},

(20, 38},

(28, 84},

(40, 59},

(51, 67},

{54, 34},

(62, 19},

(63, 81},

(64, 60},

(71, 91},

(87, 24},

(93, 73},

(95, 75},

(99, 78},

// taken from https://stackoverflow.com/a/2333816

template <template<class, class, class...> class C, typename K, typename V, typename...

»' 8 >

V GetWithDef(const C<K, V, Args...>& m, K const& key, const V & defval) {

typename C<K, V, Args...>::const\_iterator it = m.find(key); if(it m.end())

return defval; return it->second;

int turn(int player, int square) ( while (true) (

int roll = rollDice();

printf("Player %d, on square %d, rolls a %d", player, square, roll); if(square + roll > 100) {

printf(" but cannot move.\n");

} else (

square += roll;

printf(" and moves to square %d\n", square); if(square == 100) return 100;

int next = GetWithDef(snl, square, square); if(square < next) {

printf("Yay! Landed on a ladder. Climb up to %d.\n", next); square = next;

} else if(next < square) (

printf("Oops! landed on a snake. Slither down to %d.\n", next); square = next;

if(roll < 6 || !sixesThrowAgain)return square; printf("Rolled a 6 so roll again.\n");

int main() (

// three players starting on square one int players[] = ( 1, 1, 1 };

while (true) (

for (int i = 0; i < sizeof(players) / sizeof(int); ++i) ( int ns = turn(i + 1, players[i]);

if(ns == 100) {

printf("Player %d wins!\n", i + 1); goto out;

players[i] = ns; printf("\n");

out:

return 0;