Федеральное государственное автономное образовательное учреждение

Санкт-Петербургский политехнический университет Петра Великого

(ФГАОУ ВО «СПБПУ»)

Институт среднего профессионального образования

ОТЧЕТ

Практическая работа №4

О выполнении задачи

по дисциплине: «Основы программирования»

Выполнил студент: Якунин Е.А.

группы 22928/3

Преподаватель: Уцеховский А.А.

Задание #1

Написать программу моделирования бросков игральной кости.

Ввести число бросков N от 1000 до 10000, проверить правильность ввода, предоставив пользователю 5 попыток, в случае некорректного ввода 5 раз подряд, завершить программу.

По результатам испытаний (N бросков) подсчитать число выпадений n01 «1», n02 «2» … и т.д. n06 «6». Подсчитать процент выпадения каждого количества очков по формуле (n01/N)\*100, (n02/N)\*100 … и т.д. (n06/N)\*100.

Оценить равномерность случайной величины, подсчитав разности по модулю процентов выпадения каждого числа очков со средним процентом (1/6)\*100, и просуммировав эти результаты.

Программу отработать для двух случаев корректного ввода и одного некорректного.

Алгоритм решения:

Пользователь вводит с клавиатуры количество бросков, мы это число запоминаем

Цикл с условием проверки ввода кол-ва попыток, а также кол-ва попыток бросков

Следующий цикл: делаю перебор выпадений. В зависимости от рандомно выбранного числа, прибавляется определенный элемент массива.

Далее считается процент выпадения и оценивается равномерность случайной величины.

Вывод на консоль с помощью cout и printf

Код:

#include <iostream>

#include <math.h>

#include <cmath>

#include <stdio.h>

#include <ctime>

#include <iomanip>

using namespace std;

int main(){

srand(time(NULL));

double procent, deviation;

int random, throws;

double sum = 0.0;

int att = 4;

int Num = 1;

const int N = 6;

int cube[N] = {0,0,0,0,0,0};

cout << "Enter the amount of throws => ";

cin >> throws;

cout << endl;

//Throw quontity < 1000

for (int i = 1; i <= 5; i++) {

if (throws < 1000 || throws > 10000) {

cout << "Invalid input, number should be at leaset 1000. Please enter correct number. Attempts left: " << att << endl;

att--;

if (att < 0) {

cout << "Program was stopped due to invalid input." << endl;

return 0;

}

else {

cout << "Enter the amount of throws => ";

cin >> throws;

cout << endl;

}

}

}

//Header output

cout << " № " << setw(19) << "Drop counter" << setw(26) << "Procent portion" << setw(27) << "Deviation" << endl;

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

random = rand() % 6;

cube[random]++;

}

//Result output loop

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

procent = ((double)cube[i] / (double)throws) \* 100.0;

deviation = abs(procent - (100.0 / 6.0));

sum += deviation;

cout << " " << Num++ << setw(15) << cube[i] << setw(20) << " ";

printf("%.3f", procent);

cout << setw(25) << " ";

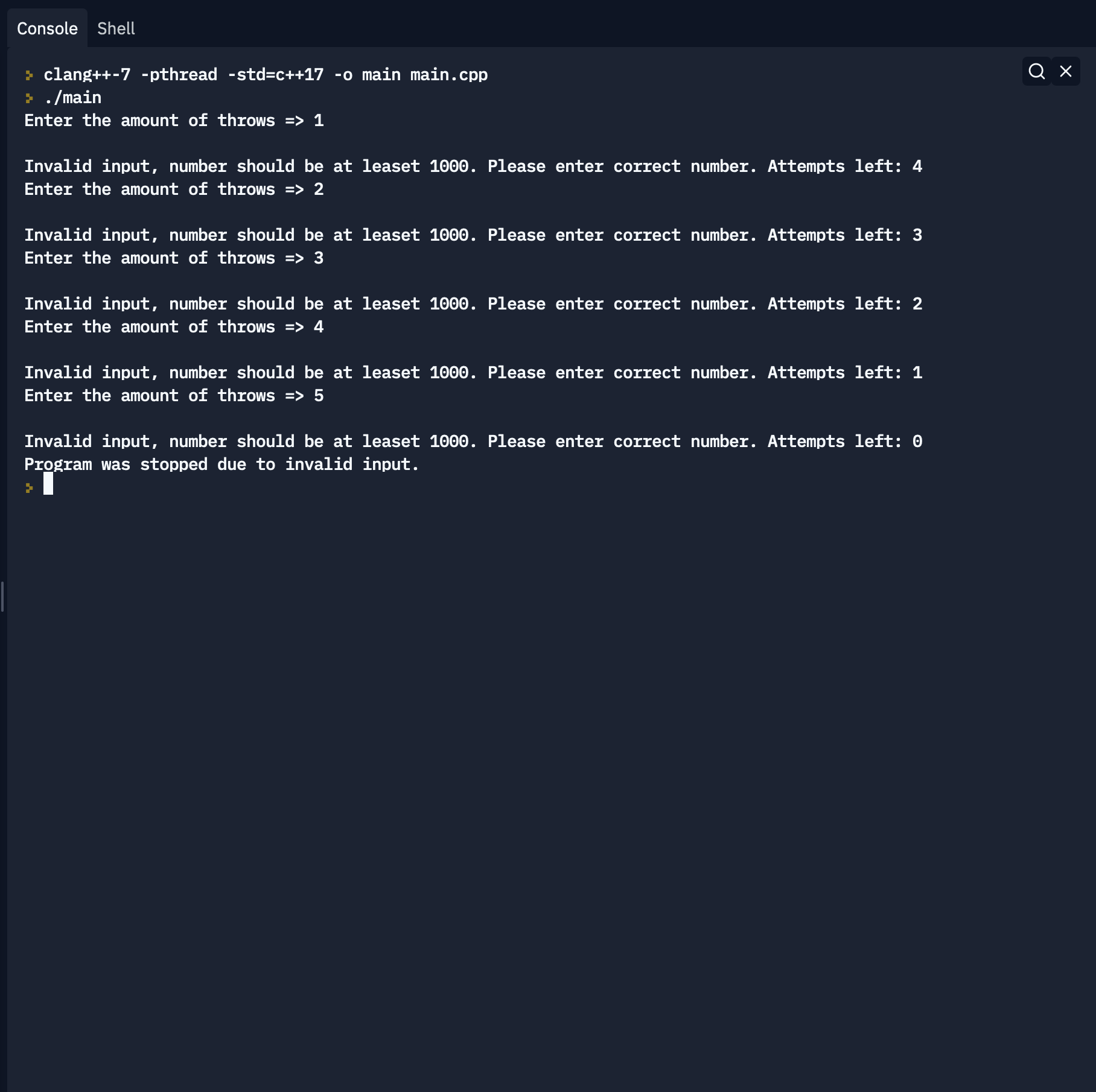
printf("%.3f\n", deviation);

}

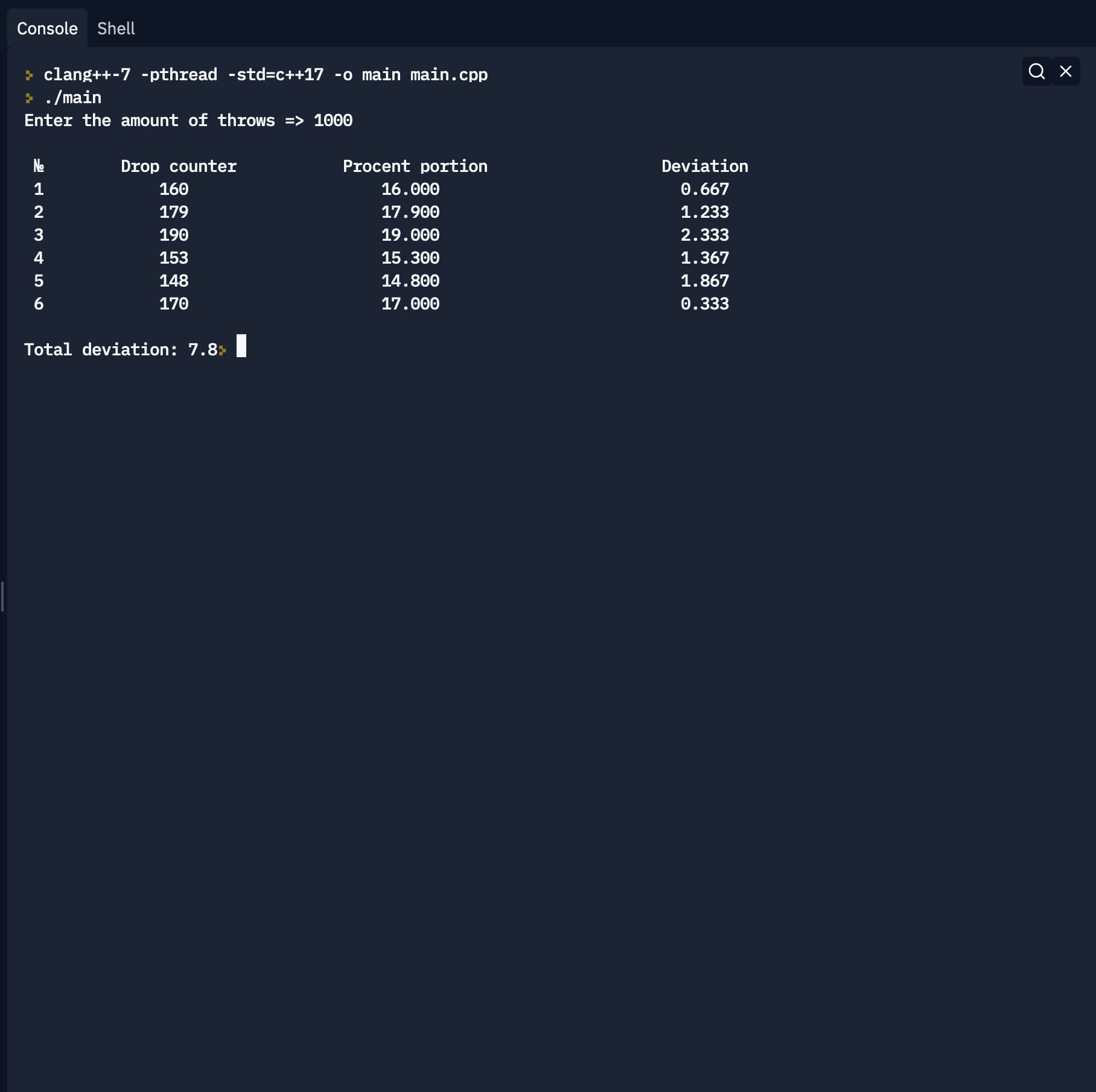
cout << "\nTotal deviation: " << sum;

}

Результаты выполнения программы:

Ошибка

Корректный ввод



Задание #2

Написать программу для моделирования игры двух игроков в кости.

Каждый игрок бросает две игральные кости, выпавшие очки суммируются. Игра состоит из трех бросков для каждого игрока. Побеждает тот, у кого по результатам бросков получается наибольшее количество очков.

Ничья означает один зачетный балл, победа дает два зачетных балл а, проигрыш – ноль. Игру и присвоение очков по ее результатам оформить отдельной функцией. Серия (или сет) состоит из пяти игр.

В результате моделирования для каждой игры вывести сумму очков по каждому из игроков, определить победителя и присвоить соответствующие зачетные баллы. По результатам сета привести суммы зачетных баллов и определить победителя. Провести две серии игр в кости.

Алгоритм решения:

Главный цикл с сетами(их 3)

Следующий цикл с играми (их 5 в одном сете)

Есть 2 массива для игрока А и для игрока B

Заполняю массив рандомным значением выпадения кубиков для игрока А и B

Инициализация и вывод результатов

Через if проверяю, какой из игроков победил

Код:

#include <iostream>

#include <math.h>

#include <cmath>

#include <ctime>

#include <iomanip>

using namespace std;

int main() {

srand(time(NULL));

int RandNum, pointsA = 0, pointsB = 0, NumGame = 1, Games = 5;

int Player = 1;

const int N = 6;

//Set loop

for (int Set = 1; Set < 3; Set++) {

cout << " Set №" << Set << endl;

cout << endl;

NumGame = 1;

//Game loop

for (int j = 0; j < Games; j++) {

int sumA = 0, sumB = 0;

int ArrPlayerA[N] = { 0,0,0,0,0,0 };

int ArrPlayerB[N] = { 0,0,0,0,0,0 };

//Loop for player's A cube random value

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

RandNum = rand() % 6;

ArrPlayerA[RandNum] += RandNum + 1;

}

//Loop for player's B cube random value

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

RandNum = rand() % 6;

ArrPlayerB[RandNum] += RandNum + 1;

}

//Result loop

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

{

sumA += ArrPlayerA[i];

sumB += ArrPlayerB[i];

}

cout << " Game №" << NumGame << endl;

cout << " Points A = " << sumA << endl;

cout << " Points B = " << sumB << endl;

if (sumA > sumB) {

pointsA += 2;

pointsB += 0;

cout << " Player A Wins! Game №" << NumGame++ << " completed with points: A = " << pointsA << " B = " << pointsB << endl;

}

else if (sumA < sumB) {

pointsA += 0;

pointsB += 2;

cout << " Player B Wins! Game №" << NumGame++ << " completed with points: A = " << pointsA << " B = " << pointsB << endl;

}

else {

pointsA += 1;

pointsB += 1;

cout << " Draw! Game №" << NumGame++ << " completed with points: A = " << pointsA << " B = " << pointsB << endl;

}

cout << endl;

}

cout << " Set №" << Set << " completed with points: A = " << pointsA << " B = " << pointsB << endl;

cout << endl;

cout << " Game total points: A = " << pointsA << " B = " << pointsB << endl;

cout << endl;

}

// Winner output

if (pointsA > pointsB) {

cout << "\nPlayer A wins!";

}

else if (pointsB > pointsA) {

cout << "\nPlayer B wins!";

}

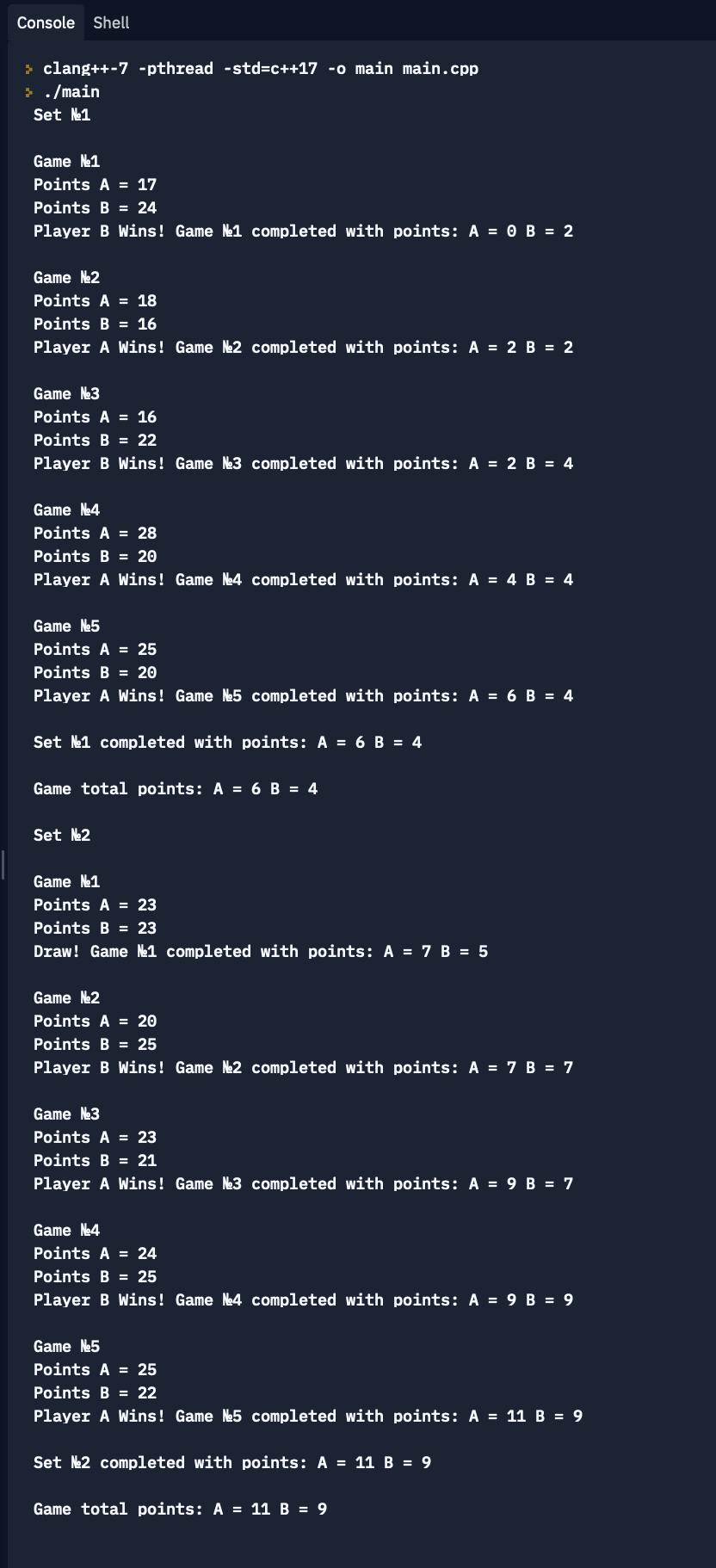
else {

cout << "\nDraw";

}

cout << endl;

}

Результаты выполнения программы: