//Exno\_2.3a

#include <iostream>

#include <string>

using std::cin; using std::endl;

using std::cout; using std::string;

int main()

{

std::cout << "Please enter P1 name: ";

std::string p1\_name;

std::cin >> p1\_name;

std::cout << "Please enter P2 name: ";

std::string p2\_name;

std::cin >> p2\_name;

const std::string player1 = "Player 1: " + p1\_name;

const std::string player2 = "Player 2: " + p2\_name;

std::cout << player1 << std::endl;

std::cout << player2 << std::endl;

constexpr int pad = 1;

constexpr int rows = pad \* 2 + 3;

const string::size\_type cols = player1.size() + pad \* 2 + 2;

cout << endl;

for (int r = 0; r != rows; ++r) {

string::size\_type c = 0;

while (c != cols) {

if (r == pad + 1 && c == pad + 1)

{

cout << player1;

c += player1.size();

}

else

{

if (r == 0 && c == 0 || r == 0 && c == player1.size() + 3

|| r == rows-1 && c== 0 || r == rows - 1 && c == player1.size()+ 3)

cout<< "\*";

else if (c == 0 || c == player1.size() + 3)

cout << "\*";

else if (r == 0 || r == rows - 1 ||

c == 0 || c == cols - 1 )

cout << "\*";

else

cout << " ";

++c;

}

}

cout << endl;

}

constexpr int pad1 = 1;

constexpr int rows1 = pad1 \* 2 + 3;

const string::size\_type cols1 = player1.size() + pad1 \* 2 + 2;

for (int r = 1; r != rows1; ++r) {

string::size\_type c = 0;

while (c != cols1) {

if (r == pad1 + 1 && c == pad1 + 1)

{

cout << player2;

c += player2.size();

}

else

{

if (r == rows - 1 && c == 0 || r == rows - 1 && c == player1.size() + 3)

cout << "\*";

else if (c == 0 || c == player1.size() + 3)

cout << "\*";

else if (r == rows1 - 1 ||

c == 0 || c == cols1 - 1 )

cout << "\*";

else

cout << " ";

++c;

}

}

cout << endl;

}

return 0;

}

==================================

//Exno\_2.3b

#include <iostream>

#include <string>

using std::cin; using std::endl;

using std::cout; using std::string;

int main()

{

std::cout << "Please enter P1 name: ";

std::string p1\_name;

std::cin >> p1\_name;

std::cout << "Please enter P2 name: ";

std::string p2\_name;

std::cin >> p2\_name;

const std::string player1 = "Player 1: " + p1\_name;

const std::string player2 = "Player 2: " + p2\_name;

std::cout << player1 << std::endl;

std::cout << player2 << std::endl;

constexpr int pad = 1;

constexpr int rows = pad \* 2 + 3;

const string::size\_type cols = player1.size() + pad \* 2 + 2;

cout << endl;

for (int r = 0; r != rows; ++r) {

string::size\_type c = 0;

while (c != cols) {

if (r == pad + 1 && c == pad + 1)

{

cout << player1;

c += player1.size();

}

else

{

if (r == 0 && c == 0 || r == 0 && c == player1.size() + 3

|| r == rows-1 && c== 0 || r == rows - 1 && c == player1.size()+ 3)

cout<< "+";

else if (c == 0 || c == player1.size() + 3)

cout << "|";

else if (r == 0 || r == rows - 1 ||

c == 0 || c == cols - 1 )

cout << "-";

else

cout << " ";

++c;

}

}

cout << endl;

}

constexpr int pad1 = 1;

constexpr int rows1 = pad1 \* 2 + 3;

const string::size\_type cols1 = player1.size() + pad1 \* 2 + 2;

for (int r = 1; r != rows1; ++r) {

string::size\_type c = 0;

while (c != cols1) {

if (r == pad1 + 1 && c == pad1 + 1)

{

cout << player2;

c += player2.size();

}

else

{

if (r == rows - 1 && c == 0 || r == rows - 1 && c == player1.size() + 3)

cout << "+";

else if (c == 0 || c == player1.size() + 3)

cout << "|";

else if (r == rows1 - 1 ||

c == 0 || c == cols1 - 1 )

cout << "-";

else

cout << " ";

++c;

}

}

cout << endl;

}

return 0;

}

=====================================

//Ex3.1

/\*#include<iostream>

using namespace std;

int main()

{

int a = 0;

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

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

std::cout<<"\*";

}

std::cout<<"\n";

}

return 0;

}

\*/

//Ex3.2

#iclude<iostream>

using namespace std;

int main()

{

int a = 4;

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

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

std::cout<<"\*";

}

std::cout<<"\n";

}

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

for(int j = a; j >= i; j--){

std::cout<<"\*";

}

std::cout<<"\n";

}

return 0;

}

==============================

#Ex3\_3

#include<iostream>

using namespace std;

int main()

{

int a = 4;

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

for(int j = 0; j < a - i; j++){

std::cout<<" ";

}

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

std::cout<<"\*";

}

std::cout<<"\n";

}

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

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

std::cout<<" ";

}

for(int k=a; k>=i;k--){

std::cout<<"\*";

}

std::cout<<"\n";

}

return 0;

}

===========================

//Exno2.3c

#include <iostream>

#include <string>

using std::cin; using std::endl;

using std::cout; using std::string;

int main()

{

std::cout << "Please enter P1 name: ";

std::string p1\_name;

std::cin >> p1\_name;

std::cout << "Please enter P2 name: ";

std::string p2\_name;

std::cin >> p2\_name;

const std::string player1 = "Player 1: " + p1\_name;

const std::string player2 = "Player 2: " + p2\_name;

std::cout << player1 << std::endl;

std::cout << player2 << std::endl;

constexpr int pad = 1;

constexpr int rows = pad \* 2 + 3;

const string::size\_type cols = player1.size() + pad \* 2 + 2;

cout << endl;

for (int r = 0; r != rows; ++r) {

string::size\_type c = 0;

while (c != cols) {

if (r == pad + 1 && c == pad + 1)

{

cout << player1;

c += player1.size();

}

else

{

if (r == 0 && c == 0 || r == 0 && c == player1.size() + 3

|| r == rows-1 && c== 0 || r == rows - 1 && c == player1.size()+ 3)

cout<< "+";

else if (r == rows - 1 && c != 0 || r == rows - 1 && c != player1.size() + 3)

cout << "-";

else if (c == 0 || c == player1.size() + 3)

cout << "|";

else if (r == 0 || r == rows - 1 ||

c == 0 || c == cols - 1 )

cout << "=";

else

cout << " ";

++c;

}

}

cout << endl;

}

constexpr int pad1 = 1;

constexpr int rows1 = pad1 \* 2 + 3;

const string::size\_type cols1 = player1.size() + pad1 \* 2 + 2;

for (int r = 1; r != rows1; ++r) {

string::size\_type c = 0;

while (c != cols1) {

if (r == pad1 + 1 && c == pad1 + 1)

{

cout << player2;

c += player2.size();

}

else

{

if (r == rows - 1 && c == 0 || r == rows - 1 && c == player1.size() + 3)

cout << "+";

else if (c == 0 || c == player1.size() + 3)

cout << "|";

else if (r == rows1 - 1 ||

c == 0 || c == cols1 - 1 )

cout << "=";

else

cout << " ";

++c;

}

}

cout << endl;

}

return 0;

}

==========================================

//Exno\_2.2

#include <iostream>

#include <string>

using std::cin; using std::endl;

using std::cout; using std::string;

int main()

{

std::cout << "Please enter P1 name: ";

std::string p1\_name;

std::cin >> p1\_name;

std::cout << "Please enter P2 name: ";

std::string p2\_name;

std::cin >> p2\_name;

const std::string player1 = "Player 1: " + p1\_name + "\*";

const std::string player2 = "Player 2: " + p2\_name;

std::cout << player1 << std::endl;

std::cout << player2 << std::endl;

constexpr int pad = 1;

constexpr int rows = pad \* 2 + 3;

const string::size\_type cols = player1.size() + player2.size() + pad \* 2 + 2;

cout << endl;

for (int r = 0; r != rows; ++r) {

string::size\_type c = 0;

while (c != cols) {

if (r == pad + 1 && c == pad + 1)

{

cout << player1;

cout << player2;

c += player1.size();

}

else

{

if (r == 0 && c == 0 || r == 0 && c == player1.size() + 3

|| r == rows-1 && c== 0 || r == rows - 1 && c == player1.size()+ 3)

cout<< "\*";

else if (c == 0 || c == player1.size() + 3)

cout << "\*";

else if (r == 0 || r == rows - 1 ||

c == 0 || c == cols - 1 )

cout << "\*";

else

cout << " ";

++c;

}

}

cout << endl;

}

return 0;

}

( OR ) MPPK

#include <iostream>

#include <string>

// say what standard-library names we use

using std::cin; using std::endl;

using std::cout; using std::string;

int main()

{

// ask for the person's name

cout << "Please enter your first name: ";

// read the name

string name;

cin >> name;

cout << "Please enter your second name: ";

string name1;

cin >> name1;

// build the message that we intend to write

const string player = "Hello, " + name + "!";\

const string player1 = " \* Hello, " + name1 + "!";\

// the number of blanks surrounding the greeting

constexpr int pad = 1;

// the number of rows and columns to write

constexpr int rows = pad \* 2 + 3;

const string::size\_type cols = player.size() + player1.size()+ pad \* 3 + 2;

// write a blank line to separate the output from the input

cout << endl;

for (int r = 0; r != rows; ++r) {

string::size\_type c = 0;

// invariant: we have written `c' characters

// so far in the current row

while (c != cols) {

// is it time to write the greeting?

if (r == pad + 1 && c == pad + 1 ) {

cout << player;

cout<< player1;

c += player.size() + player1.size();

}

else {

// are we on the border?

if (r == 0 || r == rows - 1 ||

c == 0 || c == cols - 1 || c == cols / 2 || r == 0)

cout << "\*";

else

cout << " ";

++c;

}

}

cout << endl;

}

return 0;

}

========================================

//// random.hpp

#ifndef MY\_RANDOM\_HPP

#define MY\_RANDOM\_HPP

#include <random>

class Rand\_double {

public:

using seed\_type = std::random\_device::result\_type;

Rand\_double(double low, double high): dist{low,high} {}

// draw an integer number

double operator()() { return dist(re); }

// choose new random engine seed

void seed(seed\_type s) { re.seed(s); }

private:

std::default\_random\_engine re;

std::uniform\_real\_distribution<double> dist;

};

#include <iomanip>

#include <iostream>

#include <vector>

#include <math.h>

template<typename T\_>

inline constexpr

T\_ pi\_v{3.141592653589793238462643383279502884L};

inline constexpr double pi = pi\_v<double>;

int main()

{

constexpr double rnd\_min = -1.0, rnd\_max = 1.0;

Rand\_double rnd{rnd\_min, rnd\_max};

std::random\_device rd;

rnd.seed(rd());

std::cout << std::fixed << std::setprecision(3);

/\*

double x1 = rnd();

double y1 = rnd();

std::cout << "Point #1: (" << x1 << ", " << y1 << ")\n";

double x2 = rnd();

double y2 = rnd();

std::cout << "Point #2: (" << x2 << ", " << y2 << ")";

std::cout << std::endl;

\*/

int N = 1000;

int r = 0;

int number\_Ni = 0;

while (r != N) {

double x3 = rnd(); double y3 = rnd();

double d = sqrt(abs(x3)\*abs(x3) + abs(y3) \*abs(y3));

//std::cout << "Point: (" << x3 << ", " << y3 << ")\n";

//std::cout << d;

if (d > -1 && d < 1)

++number\_Ni;

++r;

}

double probability = number\_Ni;

double estimate =( probability \* 4) / N;

std::cout << estimate << std::endl;

double relative\_error = M\_PI - estimate;

double percent\_error = (relative\_error \* 100) / M\_PI;

std::cout << "Relative error: "<< relative\_error << std::endl;

std::cout << "Relative percent error: "<< percent\_error << std::endl;

return 0;

}

// end::lab1-3b[]

#endif /\* MY\_RANDOM\_HPP \*/

=============================

Lab Hw .

1.1 above

#include <iostream>

#include <string>

// say what standard-library names we use

using std::cin; using std::endl;

using std::cout; using std::string;

int main()

{

// ask for the person's name

cout << "Please enter your first name: ";

// read the name

string name;

cin >> name;

cout << "Please enter your second name: ";

string name1;

cin >> name1;

cout << "Please enter your first name: ";

// read the name

string name2;

cin >> name2;

cout << "Please enter your second name: ";

string name3;

cin >> name3;

// build the message that we intend to write

const string player = "Hello, " + name + "!";\

const string player1 = " Hello, " + name1 + "!";\

// the number of blanks surrounding the greeting

constexpr int pad = 1;

// the number of rows and columns to write

constexpr int rows = pad \* 2 + 3;

const string::size\_type cols = player.size() + player1.size()+ pad \* 3 + 2;

// write a blank line to separate the output from the input

cout << endl;

for (int r = 0; r != rows; ++r) {

string::size\_type c = 0;

// invariant: we have written `c' characters

// so far in the current row

while (c != cols) {

// is it time to write the greeting?

if (r == pad + 1 && c == pad + 1 ) {

cout << player;

cout<< player1;

c += player.size() + player1.size();

}

else {

// are we on the border?

if (r == 0 || r == rows - 1 ||

c == 0 || c == cols - 1 || c == ((cols / 2) - 1) )

cout << "\*";

else

cout << " ";

++c;

}

}

cout << endl;

}

return 0;

}

=================