PISTAS.

Pista 1: Comience importando las bibliotecas necesarias para su programa, incluyendo **<iostream>** para la entrada/salida en la consola, **<cmath>** para la función pow() y **<numbers>** para la constante pi. Agregue un **using namespace std;** para evitar tener que usar el prefijo **std::** en cada línea.

1. #include <iostream> #include <cmath> #include <numbers> using namespace std;

Pista 2: En la tarea 1, debe definir una lambda que calcule el área de un círculo dado su radio. La lambda debe tomar un parámetro **double** y devolver un **double**. Debe utilizar la constante pi de la biblioteca **<numbers>** para realizar el cálculo.

1. auto lambdaAreaCirculo = [](double r) { return std::numbers::pi \* r \* r; };

Pista 3: En la tarea 2, debe definir una lambda genérica que devuelva el máximo de dos valores. La lambda debe tomar dos parámetros de tipo auto y devolver el tipo de parámetro común. Utilice el operador ternario para comparar los valores y devolver el máximo.

1. auto lambdaMaximo = [](auto a, auto b) { return a > b ? a : b; };

Pista 4: En la tarea 3, debe definir una plantilla lambda que calcule la potencia de un número dado una base y un exponente. La plantilla debe tomar dos parámetros de tipo T y devolver un valor de tipo T. Debe utilizar la función **pow()** de la biblioteca **<cmath>** para realizar el cálculo.

1. auto lambdaPotencia = []<typename T>(T base, T exponente) { return pow(base, exponente); };

Pista 5: Finalmente, puede completar el programa utilizando las lambdas creadas en las tareas anteriores. Ejecute cada lambda con los valores de entrada adecuados y muestre el resultado en la consola.

1. int main() {

2. // Tarea 1: Lambda que calcula el área de un círculo dado su radio.

3. auto lambdaAreaCirculo = [](double r) {

4. return std::numbers::pi \* r \* r;

5. };

6. std::cout << "El área del círculo con radio 5.0 es " << lambdaAreaCirculo(5.0) << std::endl;

7. std::cout << "El área del círculo con radio 10.0 es " << lambdaAreaCirculo(10.0) << std::endl;

8.

9. // Tarea 2: Lambda genérica que devuelve el máximo de dos valores.

10. auto lambdaMaximo = [](auto a, auto b) {

11. return a > b ? a : b;

12. };

13. std::cout << "El máximo entre 2 y 3 es " << lambdaMaximo(2, 3) << std::endl;

14. std::cout << "El máximo entre 7.5 y 4.2 es " << lambdaMaximo(7.5, 4.2) << std::endl;

15.

16. // Tarea 3: Plantilla lambda que calcula la potencia de un número dado una base y un exponente.

17. auto lambdaPotencia = []<typename T>(T base, T exponente) { return pow(base, exponente); }; std::cout << "2 elevado a 3 es " << lambdaPotencia(2, 3) << std::endl; std::cout << "5.0 elevado a 2 es " << lambdaPotencia(5.0, 2.0) << std::endl;

18. return 0; }

19.