Gonzalez Rhonal – Recuperatorio parcial 1

/\*

Ejercicio 1

Un grupo de estudiantes de geometría necesita de una aplicación que le permita obtener a partir de ciertas medidas

ingresadas el perímetro y área de distintos tipos de triángulos.

La aplicación debe permitir el ingreso de las longitudes de los triángulos hasta que se ingrese la palabra “FIN”.

La aplicación debe permitir calcular e informar:

a. El tipo de cada triángulo (isósceles, escaleno, equilátero)

b. El perímetro de cada triángulo

c. El área de cada triángulo. Recuerde que puede utilizar la fórmula de Herón: A = √(s(s − a)(s − b)(s − c)

i. Dónde:

1. s es el semiperímetro del triángulo

2. a, b y c son los lados del triángulo.

d. Obtener la posición, y el perímetro del triángulo de mayor perímetro ingresado.

Se pide:

1. Para los puntos a, b y c desarrolle procedimientos o funciones según considere más apropiado.

2. Diagrama de flujo de la función main.

usar sqrt()

semiperímetro: la mitad del perímetro.

Entradas:

- Lado1, Lado2, Lado3 hasta que se ingrese FIN

Salidas:

- tipoTriangulo

- perímetro

- areaTriangulo

\*/

#include <iostream>

#include <math.h>

using namespace std;

void leerFloat(string mensaje, float &valor);

void leerString(string mensaje, string &valor);

void obtenerLados(float &lado1, float &lado2, float &lado3);

bool esTriangulo(float lado1, float lado2, float lado3);

//a

string obtenerTipoTriangulo(float lado1, float lado2, float lado3);

//b

float calcularPerimetro(float lado1, float lado2, float lado3);

//c

float calcularAreaTriangulo(float semiperimetro, float lado1, float lado2, float lado3);

//d

void verificarMayorPerimetro(float perimetro, float &mayorPerimetro, int posicion, int &posicionMayorPerimetro);

int main()

{

//Entradas

float lado1, lado2, lado3;

//Salidas

string tipoTriangulo;

float perimetro, areaTriangulo;

int posicionMayorPerimetro = 0;

float mayorPerimetro = 0;

string opcion;

float semiperimetro;

int posicion = 0;

leerString("¿Desea ingresar un triángulo? ('S' para sí o 'FIN' para salir): ", opcion);

while (opcion != "FIN")

{

obtenerLados(lado1, lado2, lado3);

if (esTriangulo(lado1, lado2, lado3))

{

posicion++;

tipoTriangulo = obtenerTipoTriangulo(lado1, lado2, lado3);

perimetro = calcularPerimetro(lado1, lado2, lado3);

verificarMayorPerimetro(perimetro, mayorPerimetro, posicion, posicionMayorPerimetro);

semiperimetro = perimetro / 2;

areaTriangulo = calcularAreaTriangulo(semiperimetro, lado1, lado2, lado3);

cout << "Ingresó un triángulo " << tipoTriangulo << endl;

cout << "Su perímetro es: " << perimetro << endl;

cout << "Y su área es: " << areaTriangulo << endl

<< endl;

}

else

{

cout << "Los lados ingresados no forman un triángulo." << endl

<< endl;

}

leerString("¿Desea ingresar otro triángulo? ('S' para sí o 'FIN' para salir): ", opcion);

}

//d. Obtener la posición, y el perímetro del triángulo de mayor perímetro ingresado.

if (posicion == 0)

{

cout << "No se ingresaron triángulos válidos." << endl;

}

else

{

cout << "El mayor perímetro ingresado fue de " << mayorPerimetro << endl

<< " y corresponde al triángulo váĺido ingresado en la posición " << posicionMayorPerimetro << endl

<< endl;

}

return 0;

}

void obtenerLados(float &lado1, float &lado2, float &lado3)

{

leerFloat("Ingrese el primer lado: ", lado1);

leerFloat("Ingrese el segundo lado: ", lado2);

leerFloat("Ingrese el tercer lado: ", lado3);

}

void leerFloat(string mensaje, float &valor)

{

cout << mensaje;

cin >> valor;

cout << endl;

return;

}

void leerString(string mensaje, string &valor)

{

cout << mensaje;

cin >> valor;

cout << endl;

return;

}

bool esTriangulo(float lado1, float lado2, float lado3)

{

return lado1 + lado2 > lado3 && lado1 + lado3 > lado2 && lado2 + lado3 > lado1;

}

//a. El tipo de cada triángulo (isósceles, escaleno, equilátero)

string obtenerTipoTriangulo(float lado1, float lado2, float lado3)

{

string tipoTriangulo = "";

if (lado1 == lado2 && lado2 == lado3)

{

tipoTriangulo = "Equilatero";

}

else

{

if (lado1 != lado2 && lado2 != lado3 && lado1 != lado3)

{

tipoTriangulo = "Escaleno";

}

else

{

tipoTriangulo = "Isosceles";

}

}

return tipoTriangulo;

}

//b. El perímetro de cada triángulo

float calcularPerimetro(float lado1, float lado2, float lado3)

{

return lado1 + lado2 + lado3;

}

//c. El área de cada triángulo. Recuerde que puede utilizar la fórmula de Herón: A = √(s(s − a)(s − b)(s − c))

float calcularAreaTriangulo(float semiperimetro, float lado1, float lado2, float lado3)

{

float radicando = semiperimetro \* (semiperimetro - lado1) \* (semiperimetro - lado2) \* (semiperimetro - lado3);

return sqrt(radicando); //√(s(s − a)(s − b)(s − c))

}

//d

void verificarMayorPerimetro(float perimetro, float &mayorPerimetro, int posicion, int &posicionMayorPerimetro)

{

if (perimetro > mayorPerimetro || mayorPerimetro == 0)

{

mayorPerimetro = perimetro;

posicionMayorPerimetro = posicion;

}

return;

}

/\*

Ejercicio 2

Se desea obtener una aplicación que permita convertir números binarios en un números decimales. Para ello la aplicación

deberá permitir el ingreso de números hasta que el usuario ingrese “FIN”. Para indicar el fin de un número binario se

ingresará el número -1.

Ejemplo, si el usuario ingresa: 1|0|0|0|1|-1 debe devolver 17 ya que:

1 x 2 0 + 0 x 2 1 + 0 x 2 2 + 0 x 2 3 + 1 x 2 4 = 1 + 0 + 0 + 0 + 16 = 17

Se pide:

a.

b.

Mostrar cada número decimal obtenido.

Mostrar la posición y el número decimal del menor número decimal obtenido.

\*/

#include <iostream>

#include <math.h>

using namespace std;

void leerString(string mensaje, string &valor)

{

cout << mensaje;

cin >> valor;

cout << endl;

return;

}

void leerInt(string mensaje, int &valor)

{

cout << mensaje;

cin >> valor;

cout << endl;

return;

}

int main()

{

string opcion;

int numero;

int digitoDecimal;

int numeroDecimal = 0;

int cantNumeros;

int menorNumeroDecimal = 0;

int posicion = 0, posicionMenor;

leerString("Desea ingresar un número binario? ('S' para sí o 'FIN' para salir) ", opcion);

while (opcion != "FIN")

{

posicion++;

leerInt("Ingrese un numero entre 0 y 1 o -1 para salir", numero);

while (numero >= 0)

{

digitoDecimal = numero \* pow((double)2, (double)cantNumeros);

numeroDecimal += digitoDecimal;

cantNumeros++;

leerInt("Ingrese un numero entre 0 y 1 o -1 para salir", numero);

}

cout << "El número es: " << numeroDecimal << endl;

if (numeroDecimal < menorNumeroDecimal || menorNumeroDecimal == 0)

{

menorNumeroDecimal = numeroDecimal;

posicionMenor = posicion;

}

numeroDecimal = 0;

cantNumeros = 0;

}

cout << "El menor numero decimal fue " << menorNumeroDecimal << endl

<< "Fue ingresado en la posicion " << posicionMenor << endl

<< endl;

return 0;

}