Nome: Barbara Diogo.

Matrícula: 2063853.

Nome: Felipe de Araujo Teixeira.

Matrícula: 2046950.

Exercício 1: Utilizando as series de potência do seno e cosseno, pede-se que imprima os resultados para 3 termos (n=0 até n=2), logo em seguida para 40 termos (n=0 até n=39). Foi solicitado também a biblioteca cmath para as funções sin() e cos() e o erro percentual.

Na resolução, separamos em funções, uma para calcular o fatorial, 2 para o seno (mudando apenas o número de termos) e 2 para o cosseno (mudando apenas o número de termos). Desta forma, deixamos como opção para o usuário escolher para quantos termos o cálculo seria feito e a partir de sua escolha determinaria qual das funções seria utilizada.

Para facilitar a digitação do usuário, optamos em receber o angulo em graus, e dentro do main fizemos a conversão deste angulo para radiano como pede a função. O erro percentual foi separado em 2 partes, para facilitar os cálculos e a visualização do código, a primeira parte refere-se ao erro para 3 termos e na segunda parte para 40 termos.

Por fim, será impresso ao usuário o Seno sem a função, Seno com a função sin(), o valor absoluto da diferença desses 2 resultados, erro percentual do Seno, o Cosseno sem a função, Cosseno com a função cos(), o valor absoluto da diferença desses 2 resultados e o erro percentual do Cosseno.

Abaixo segue o código descrito acima:

/\*

Nome: Barbara Diogo;

Matrícula: 2063853;

Nome: Felipe de Araujo Teixeira;

Matrícula: 2046950

\*/

#include<iostream>

#include<cmath>

#include <locale>

#include <windows.h>

using namespace std;

double fatorial(int n){

double fat = 1;

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

fat = fat \* i;

}

return fat;

}

//Seno para 3 termos

double Seno1(double rad, int n=3){

double somaS = 0;

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

somaS += pow(-1,i) \* (pow(rad, 2\*i+1)/fatorial(2\*i+1));

}

return somaS;

}

//Seno para 40 termos

double Seno2(double rad, int n=40){

double somaS = 0;

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

somaS += pow(-1,i) \* (pow(rad, 2\*i+1)/(double) fatorial(2\*i+1));

}

return somaS;

}

//Cosseno para 3 termos

double Cosseno1(double rad, int n=3){

double somaC = 0;

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

somaC += pow(-1,i) \* (pow(rad, 2\*i)/(double) fatorial(2\*i));

}

return somaC;

}

//Cosseno para 40 termos

double Cosseno2(double rad, int n=40){

double somaC = 0;

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

somaC += pow(-1,i) \* (pow(rad, 2\*i)/(double) fatorial(2\*i));

}

return somaC;

}

int main(){

setlocale(LC\_ALL, "pt\_BR.UTF-8");

SetConsoleOutputCP(CP\_UTF8);

int op;

double x;

double valorabsoluto, valorabsoluto2, valorabsoluto3, valorabsoluto4, erroSeno1, erroCos1;

double erroSeno2, erroCos2;

double conversao; // graus p radiano.

double pi = M\_PI; //biblioteca define várias constantes matemáticas.

cout << "\*\*\*\*\*\*\*\*\*\*\*\*"<< endl;

cout << " Calculadora Seno e Cosseno "<< endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*"<< endl;

cout << "Digite o valor do angulo(em graus): ";

cin >> x;

cout << endl << "Deseja que o calculo seja feito para: "<< endl;

cout << "1: 3 termos" << endl;

cout << "2: 40 termos"<< endl;

cout << "Opção 1 ou 2: ";

cin >> op;

cout << endl;

conversao = ((x\*pi)/180);

//valor absoluto

valorabsoluto = Seno1(conversao) - sin(conversao);

valorabsoluto2 = Cosseno1(conversao) - cos(conversao);

valorabsoluto3 = Seno2(conversao) - sin(conversao);

valorabsoluto4 = cos(conversao) - Cosseno2(conversao);

//erro percentual = [valor experimental - valor teórico] / valor teórico x 100%

erroSeno1 = (Seno1(conversao) - sin(conversao)) / (sin(conversao)\*100);

erroCos1 = (Cosseno1(conversao) - cos(conversao)) / (cos(conversao)\*100);

erroSeno2 = (Seno2(conversao) - sin(conversao)) / (sin(conversao)\*100);

erroCos2= (cos(conversao) - Cosseno2(conversao)) / (Cosseno2(conversao)\*100);

if(op == 1){

cout << "A conversao é:" <<conversao << endl;

cout << "Seno " << x << "° = " << sin(conversao) << " (COM a função sin())" << endl;

cout << "Seno " << x << "° = " << Seno1(conversao) << " (SEM a função sin())" << endl;

cout << "Valor Absoluto: " << valorabsoluto << endl;

cout << "Com erro percentual de " << erroSeno1 << "%" << endl << endl;

cout << "Cosseno " << x << "° = " << cos(conversao) << " (COM a função cos())" << endl;

cout << "Cosseno " << x << "° = " << Cosseno1(conversao) << " (SEM a função cos())" << endl;

cout << "Valor Absoluto: " << valorabsoluto2 << endl;

cout << "Com erro percentual de " << erroCos1 << "%" << endl;

}else if(op == 2){

cout << "Seno " << x << "° = " << sin(conversao) << " (COM a função sin())" << endl;

cout << "Seno " << x << "° = " << Seno2(conversao) << " (SEM a função sin())" << endl;

cout << "Valor Absoluto: " << valorabsoluto3 << endl;

cout << "Com erro percentual de " << erroSeno2 << "%" << endl << endl;

cout << "Cosseno " << x << "° = " << cos(conversao) << " (COM a função cos())" << endl;

cout << "Cosseno " << x << "° = " << Cosseno2(conversao) << " (SEM a função cos())" << endl;

cout << "Valor Absoluto: " << valorabsoluto4 << endl;

cout << "Com erro percentual de " << erroCos2 << "%" << endl << endl;

}else{

cout << "Opção inválida";

}

}

Exercício 2 – Deseja-se saber se uma determinada matriz é considera um quadrado mágico (a soma de cada linha, coluna e diagonais, devem apresentar o mesmo valor) ou não é considerada.

O código foi estruturado com diversas funcionais. A primeira dava ao usuário a possibilidade de escolha, na qual era escolhido uma matriz pronta com números sorteados automaticamente ou digitados por ele.

Em seguida, uma função que soma todos os elementos de cada linha.

A próxima função, com o mesmo objetivo, mas ao invés de somar os elementos de cada linha, soma os elementos das colunas.

Foi adicionado mais duas funções para fazer a soma dos elementos das diagonais (diagonal principal e diagonal secundária).

Para finalizar as funções, foi criada uma para a comparação da soma de todas as linhas, colunas e diagonais.

Dentro do main, todas as funções foram chamadas para que imprimissem a soma das devidas linhas, colunas e diagonais. Após isso, ocorre a comparação entra ambas somas e caso todas sejam iguais, será retornado “A matriz é considerada um quadrado mágico”, caso contrário, será retornado “A matriz não é considerada um quadrado mágico”.

Segue código referente a explicação acima:

/\*

Nome: Barbara Diogo;

Matrícula: 2063853;

Nome: Felipe de Araujo Teixeira;

Matrícula: 2046950

2- Elabore um programa que verifique se uma matriz 4x4, com números

informados pelo usuário ou gerados randomicamente (faça um menu

que suporte tal situação), forma o chamado quadrado mágico. Um

quadrado mágico é formado quando a soma dos elementos de cada

linha é igual à soma dos elementos de cada coluna dessa linha, é igual à

soma dos elementos da diagonal principal e, também, é igual à soma dos

elementos da diagonal secundária. Após a verificação, o resultado e a

matriz deverão ser impressos no console.

\*/

#include <iostream>

#include <stdlib.h>

#include <time.h>

#include <locale>

#include <windows.h>

using namespace std;

//função na qual o usuário irá escolher se deseja inserir os números da matriz ou a matriz será sorteada aleatoriamente.

void escolha(int opcao, int matriz[4][4]){

int n = 4;

cout << "Digite a opção escolhida:" << endl;

cout << "1- Matriz pronta" << endl;

cout << "2- Digitar a matriz 4x4" << endl;

cin >> opcao;

if(opcao == 1){

int i, j;

srand(time(NULL));

for(i=0; i<n; i++){

for(j=0; j<n; j++){

matriz[i][j] = rand() %100;

}

}

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

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

cout << "|" << matriz[i][j] << "|";

}

cout << endl;

}

}else{

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

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

cout << "Digite os numeros da matriz: ";

cin >> matriz[i][j];

}

}

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

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

cout << "|" << matriz[i][j] << "|";

}

cout << endl;

}

}

}

//função para o calculo da soma dos elementos das linhas da matriz.

int somaLinha(int matriz[4][4], int v[4]){

int n = 4;

int i, j;

for(i=0; i<n; i++){

v[i] = 0;

for(j=0; j<n; j++){

v[i] += matriz[i][j];

}

}

for(i=0; i<n; i++){

cout << "Soma da linha " << i << " : " << v[i] << endl;

}

return v[i];

}

//função para o calculo da soma dos elementos das colunas da matriz.

int somacoluna(int matriz[4][4], int v[4]){

int n = 4;

int i, j;

for(j=0; j<n; j++){

v[j] = 0;

for(i=0; i<n; i++){

v[j] += matriz[i][j];

}

}

for(j=0; j<n; j++){

cout << "Soma da coluna " << j << " : " << v[j] << endl;

}

return v[j];

}

//função para calculo da soma dos elementos da diagonal principal da matriz.

int somaDiagPrincipal(int x, int matriz[4][4]){

int n = 4;

int i, soma=0;

for(i=0; i<n; i++){

soma = soma + matriz[i][i];

}

return soma;

}

//função para calculo da soma dos elementos das diagonal secundária da matriz.

int somaDiagSecundaria(int x,int matriz[4][4]){

int n = 4;

int i, soma=0;

for(i=0; i<n; i++){

soma = soma + matriz[i][n-i-1];

}

return soma;

}

//função na qual verifica se a soma das linhas, colunas e diagonais apresentam o mesmo resultado;

int quadradomagico(int x, int v[4]){

int n = 4;

int i;

for(i=1; i<n; i++){

if(v[i] != v[i-1]){

return 0;

}

}

return 1;

}

int main(){

setlocale(LC\_ALL, "pt\_BR.UTF-8");

SetConsoleOutputCP(CP\_UTF8);

int x = 4;

int opcao;

int mat[4][4];

int vL[4], vC[4], testeL, testeC;

int Principal=0, Secundaria=0;

escolha(opcao, mat);

Principal = somaDiagPrincipal(x, mat);

Secundaria = somaDiagSecundaria(x, mat);

somaLinha(mat, vL);

somacoluna(mat, vC);

cout << "Soma da diagonal principal: " << Principal;

cout << endl;

cout << "Soma da diagonal secundaria: " << Secundaria;

cout << endl;

testeL = quadradomagico(x, vL);

testeC = quadradomagico(x, vC);

if (testeL && testeC && Principal == Secundaria && Principal == vL[0]){

cout << "A matriz é considerada Quadrado Magico";

}else{

cout << "A matriz não é considerada Quadrado Magico";

}

return 0;

}