Laborator 1

#include <conio.h>

// Torsor

// utilizand Dev-C++5.11 - Console Application

// Copyright © 2019 - prof. dr. ing. Andrei Craifaleanu

#include <stdio.h>

#include <math.h>

int main()

{

float a, b, c, F, M, B, C;

float X[8], Y[8], Z[8];

float Fx, Fy, Fz;

float Rx=0, Ry=0, Rz=0;

float Mx, My, Mz, Mx2, My2, Mz2;

float Mox=0, Moy=0, Moz=0;

int i, j, n, m, P, Q;

// Citirea laturilor paralelipipedului si calculul coordonatelor varfurilor

printf("Laturile paralelipipedului = ");

scanf(" %f %f %f", &a, &b, &c);

X[0] = 0; Y[0] = 0; Z[0] = 0;

X[1] = a; Y[1] = 0; Z[1] = 0;

X[2] = a; Y[2] = b; Z[2] = 0;

X[3] = 0; Y[3] = b; Z[3] = 0;

X[4] = 0; Y[4] = 0; Z[4] = c;

X[5] = a; Y[5] = 0; Z[5] = c;

X[6] = a; Y[6] = b; Z[6] = c;

X[7] = 0; Y[7] = b; Z[7] = c;

// Introducerea fortelor

printf("\n");

printf("Nr. forte = ");

scanf(" %d", &n);

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

{

printf("%d) F, P, Q = ", i);

scanf(" %f %d %d", &F, &P, &Q);

B = F/sqrt(pow(X[Q] - X[P], 2) + pow(Y[Q] - Y[P], 2) + pow(Z[Q] - Z[P], 2));

Fx =B\*(X[Q] - X[P]);

Fy = B\*(Y[Q] - Y[P]);

Fz = B\*(Z[Q] - Z[P]);

Rx += Fx;

Ry += Fy;

Rz += Fz;

Mx = Y[P]\*Fz - Z[P]\*Fy;

My = Z[P]\*Fx - X[P]\*Fz;

Mz = X[P]\*Fy - Y[P]\*Fx;

Mox += Mx;

Moy += My;

Moz += Mz;

}

printf(" Nr momente ");

scanf(" %d", &m);

for(j = 1; j <= m; j++)

{

printf("%d) M, P, Q = ", i);

scanf(" %f %d %d", &M, &P, &Q);

C = M / sqrt( pow(X[Q] - X[P], 2) + pow(Y[Q] - Y[P], 2) + pow(Z[Q] - Z[P], 2));

Mx2=C \* (X[Q] - X[P]);

My2=C \* (Y[Q] - Y[P]);

Mz2=C \* (Z[Q] - Z[P]);

Mox += Mx2;

Moy += My2;

Moz += Mz2;

}

// Afisarea rezultatelor

printf("\n");

printf("R = (%6.3f, %6.3f, %6.3f) \n", Rx, Ry, Rz);

printf("\n");

printf("R = (%6.3f, %6.3f, %6.3f) \n", Mox, Moy, Moz);

printf("\n");

printf(" Press any key! \n");

while (!kbhit()){};

return 0;

}