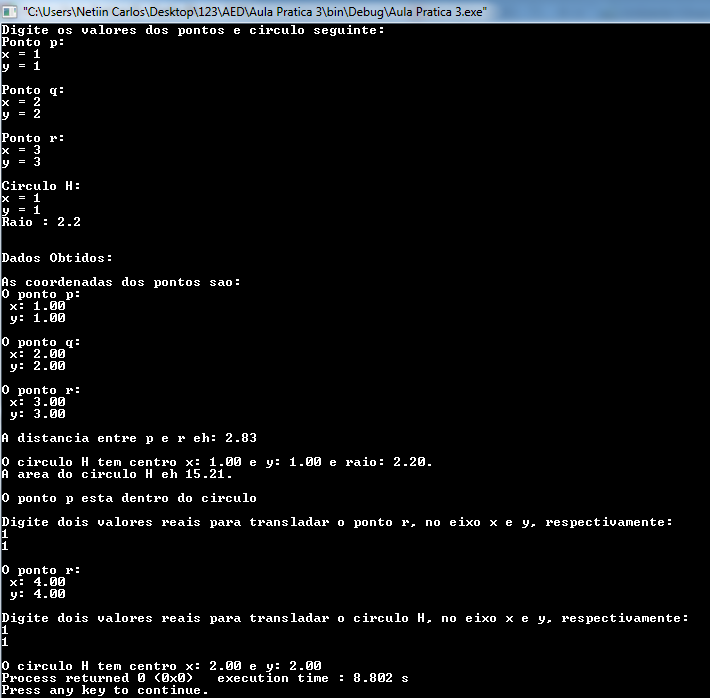
UFU

Nome: Antonio Carlos Neto

Curso: Ciência da Computação

Disciplina: GBC024

EXECUÇÃO:



main.c:

#include <stdio.h>

#include <stdlib.h>

#include "ponto.h"

#include "circulo.h"

struct ponto

{

float x;

float y;

};

struct circulo

{

Ponto\* p;

float r;

};

int main(void)

{

float a, b, c, d, r, n, m;

printf("Digite os valores dos pontos e circulo seguinte:\n");

printf("Ponto p:\nx = ");

scanf("%f",&a);

printf("y = ");

scanf("%f",&b);

Ponto\* p = cria\_Ponto(a,b);

printf("\nPonto q:\nx = ");

scanf("%f",&a);

printf("y = ");

scanf("%f",&b);

Ponto\* q = cria\_Ponto(a,b);

acessa\_Ponto(q,&c,&d);

printf("\nPonto r:\nx = ");

scanf("%f",&a);

printf("y = ");

scanf("%f",&b);

atribui\_Ponto(q,a,b);

printf("\nCirculo H:\nx = ");

scanf("%f",&a);

printf("y = ");

scanf("%f",&b);

printf("Raio : ");

scanf("%f", &r);

Circulo\* H = cria\_Circulo(a,b,r);

printf("\n\nDados Obtidos:\n\n");

printf("As coordenadas dos pontos sao:");

printf("\nO ponto p:\n x: %.2f\n y: %.2f",p->x, p->y);

printf("\n\nO ponto q:\n x: %.2f\n y: %.2f",c,d);

printf("\n\nO ponto r:\n x: %.2f\n y: %.2f",q->x, q->y);

printf("\n\nA distancia entre p e r eh: %.2f",distancia\_Pontos (p,q));

printf("\n\nO circulo H tem centro x: %.2f e y: %.2f e raio: %.2f.\nA area do circulo H eh %.2f.",a,b, r,area\_Circulo(H));

if(interior\_Circulo(H,p)==1)

{

printf("\n\nO ponto p esta dentro do circulo");

}

else if(interior\_Circulo(H,p)==0)

{

printf("\n\nO ponto p esta fora do circulo");

}

printf("\n\nDigite dois valores reais para transladar o ponto r, no eixo x e y, respectivamente:\n");

scanf("%f%f",&n,&m);

transladaPontoX(q,n);

transladaPontoY(q,m);

printf("\nO ponto r:\n x: %.2f\n y: %.2f",q->x, q->y);

printf("\n\nDigite dois valores reais para transladar o circulo H, no eixo x e y, respectivamente:\n");

scanf("%f%f",&n,&m);

transladaCirculoX (H, n);

transladaCirculoY (H, m);

printf("\nO circulo H tem centro x: %.2f e y: %.2f",(a+n),(b+m), r);

libera\_Circulo(H);

libera\_Ponto(q);

libera\_Ponto(p);

return 0;

}

ponto.c:

#include <stdlib.h>

#include <stdio.h>

#include <math.h>

#include "ponto.h"

struct ponto

{

float x;

float y;

};

Ponto\* cria\_Ponto (float x, float y)

{

Ponto\* p = (Ponto\*) malloc(sizeof(Ponto));

if (p == NULL)

{

printf("Memoria insuficiente!\n");

exit(1);

}

p->x = x;

p->y = y;

return p;

}

void libera\_Ponto (Ponto\* p)

{

free(p);

}

void acessa\_Ponto (Ponto\* p, float\* x, float\* y)

{

\*x = p->x;

\*y = p->y;

}

void atribui\_Ponto (Ponto\* p, float x, float y)

{

p->x = x;

p->y = y;

}

float distancia\_Pontos (Ponto\* p1, Ponto\* p2)

{

float dx = p2->x - p1->x;

float dy = p2->y - p1->y;

return sqrt(dx\*dx + dy\*dy);

}

void transladaPontoX(Ponto\* p, float n)

{

p->x = p->x + n;

}

void transladaPontoY(Ponto\* p, float n)

{

p->y = p->y + n;

}

ponto.h:

#ifndef PONTO\_H\_INCLUDED

#define PONTO\_H\_INCLUDED

typedef struct ponto Ponto;

Ponto\* cria\_Ponto (float x, float y);

void libera\_Ponto (Ponto\* p);

void acessa\_Ponto (Ponto\* p, float \*x, float \*y);

void atribui\_Ponto (Ponto\* p, float x, float y);

float distancia\_Pontos (Ponto\* p1, Ponto\* p2);

void transladaPontoX(Ponto\* p, float n);

void transladaPontoY(Ponto\* p, float n);

#endif // PONTO\_H\_INCLUDED

circulo.c:

#include <stdlib.h>

#include "circulo.h"

#define PI 3.14159

struct circulo

{

Ponto\* p;

float r;

};

Circulo\* cria\_Circulo(float x, float y, float r)

{

Circulo\* c= (Circulo\*)malloc(sizeof(Circulo));

c->p = cria\_Ponto(x,y);

c->r = r;

return c;

}

void libera\_Circulo(Circulo\* c)

{

libera\_Ponto(c->p);

free(c);

}

float area\_Circulo(Circulo\* c)

{

return PI\*c->r\*c->r;

}

int interior\_Circulo(Circulo\* c, Ponto\* p)

{

float d = distancia\_Pontos(c->p,p);

return (d<c->r);

}

void transladaCirculoX (Circulo\* c, float n)

{

transladaPontoX(c->p,n);

}

void transladaCirculoY (Circulo\* c, float n)

{

transladaPontoY(c->p,n);

}

circulo.h:

#ifndef CIRCULO\_H\_INCLUDED

#define CIRCULO\_H\_INCLUDED

#include "ponto.h"

typedef struct circulo Circulo;

Circulo\* cria\_Circulo(float x, float y, float r);

void libera\_Circulo (Circulo \*c);

float area\_Circulo (Circulo \*c);

int interior\_Circulo (Circulo \*c, Ponto \*p);

void transladaCirculoX (Circulo\* c, float n);

void transladaCirculoY (Circulo\* c, float n);

#endif // CIRCULO\_H\_INCLUDED