

/*Arq. ponto.h*/

```
typedef struct pto * ponto;  
  
int cria_pto (ponto *p, float x, float y);  
void libera_pto (ponto *p);  
float distancia (ponto p1, ponto p2);
```

/*Arq. ponto.c*/

```
#include <stdio.h>  
#include <math.h>  
#include "ponto.h"  
  
struct pto {float x,y};  
  
int cria_pto (ponto *p, float x, float y){  
    *p = (ponto ) malloc(sizeof(struct pto));  
    if(*p == NULL)  
        return 0; //falha na alocação do ponto  
    (*p)->x = x;  
    (*p)->y = y;  
    return 1;  
}  
void libera_pto (ponto *p){  
    free(*p);  
    *p = NULL;  
}  
float distancia (ponto p1, ponto p2){  
    float r;  
    r = sqrt(pow((p2->x - p1->x),2) + pow((p2->y - p1->y),2));  
    return r;  
}
```

/*Arq prog.c*/

```
#include <stdio.h>  
#include "ponto.h"  
main(){  
    ponto p1, p2;  
    float x, y, r;  
    puts("\nDigite as coordenadas do 1º ponto separadas por virgula:");  
    scanf("%f,%f", &x, &y);  
    if(!cria_pto(&p1, x, y)){  
        puts("\nNão foi possível criar o 1º ponto. Possivelmente a  
memória está cheia.");  
        exit(1);  
    }  
    puts("\nDigite as coordenadas do 2º ponto separadas por virgula:");  
    scanf("%f,%f", &x, &y);  
    if(!cria_pto(&p2, x, y)){  
        puts("\nNão foi possível criar o 2º ponto. Possivelmente a  
memória está cheia.");  
        exit(1);  
    }  
    r = distancia(p1,p2);  
    printf("\nA distancia entre os dois pontos eh: %.2f", r);  
  
    libera_pto(&p1);  
    libera_pto(&p2);  
}
```
