

Nome: Gabriel Couto de Freitas

Matrícula: 12021BCC040

Exercício 1:

Especificações função cria_nro:

Entrada: Nenhuma

Pré-condições: Nenhuma

Processo: Atribui e instancia endereços na memória para os números criados.

Saída: O número complexo.

Pós-Condições: Deve ser diferente de NULL.

Especificações função set_nro:

Entrada: Struct de número complexo e dois inteiros, um para a parte real e outra para imaginaria.

Pré-condições: Devem estar instanciados.

Processo: Atribui ao struct os valores inseridos na função.

Saída: 1(falha) 0(sucesso).

Pós-Condições: Nenhuma.

Especificações função libera_nro:

Entrada: Endereço de um struct.

Pré-condições: Nenhuma

Processo: Libera o endereço instanciado pelo struct.

Saída: nenhuma

Pós-Condições: nenhuma

Especificações função soma:

Entrada: Dois structs de números complexos

Pré-condições: Nenhuma

Processo: Soma a parte imaginaria e a real dos números

Saída: Retorna a soma resultante

Pós-Condições: Deve ser diferente de NULL.

Especificações função sub:

Entrada: Dois structs de números complexos

Pré-condições: nenhuma

Processo: Subtrai os dois números complexos

Saída: O resultado da subtração

Pós-Condições: Deve ser diferente de NULL.

Especificações função mult:

Entrada: Dois structs de números complexos

Pré-condições: nenhuma

Processo: Realiza a multiplicação entre dois números complexos

Saída: Retorna o produto da multiplicação

Pós-Condições: Deve ser diferente de NULL

Exercício 2:

Especificação função cria_pt:

Entrada: Três números float.

Pré-condições: nenhuma

Processo: Cria e instancia um ponto no espaço e em seguida atribui os valores inseridos

Saída: Retorna o struct

Pós-Condições: Deve ser diferente de NULL.

Especificação função `apaga_pto`:

Entrada: Endereço de um struct de ponto

Pré-condições: nenhuma

Processo: Libera os endereços instanciados do struct e preenche com NULL.

Saída: nenhuma

Pós-Condições: nenhuma

Especificação função `distancia`:

Entrada: Dois structs ponto

Pré-condições: os structs não podem ser NULL

Processo: Calcula a distância no espaço entre esses dois pontos

Saída: Retorna a distancia

Pós-Condições: deve ser maior ou igual a 0.

Exercício 3:

Especificação função `cria_esfera`:

Entrada: Três números float.

Pré-condições: nenhuma

Processo: Cria e instancia um ponto no espaço e em seguida atribui os valores inseridos

Saída: Retorna o struct

Pós-Condições: Deve ser diferente de NULL.

Especificação função `libera_esfera`:

Entrada: Endereço de um struct de ponto

Pré-condições: nenhuma

Processo: Libera os endereços instanciados do struct e preenche com NULL.

Saída: nenhuma

Pós-Condições: nenhuma

Especificação função raio:

Entrada: Dois structs de ponto

Pré-condições: Os structs não devem ser NULL

Processo: Calcula a distancia entre os dois pontos, que sera usado como raio da esfera

Saída: Retorna o raio calculado

Pós-Condições: O raio deve ser maior que 0.

Especificação função área:

Entrada: dois structs de ponto e o raio

Pré-condições: o raio deve ser maior que 0.

Processo: calcula a área da esfera de acordo com a formula

Saída: Retorna a área da esfera calculada

Pós-Condições: deve ser maior que 0

Especificação função volume:

Entrada: dois structs de ponto e o raio

Pré-condições: o raio deve ser maior que 0

Processo: calcula o volume da esfera de acordo com a formula

Saída: Retorna o volume da esfera calculada

Pós-Condições: deve ser maior que 0