

Aluna: Renata Cristina Gomes da Silva

Especificações TAD

01. a) cria - novo

- entrada: 2 valores reais, um real e um imaginário
- pré-condição: não tem
- processo: criar e atribuir um n° complexo para as partes real e imaginária
- saída: end. da memória do n° complexo com valor real e imaginário
- pós-condição: não tem

b) set - novo

- entrada: valores digitados
- pré-condição: valores reais digitados
- processo: ir modificando os valores das partes real e imaginária do n° complexo
- saída: valores modificados
- pós-condição: os valores de parte real e imaginária já modificados

c) get - novo

- entrada: end. do n° complexo, end. de um double real e end. de double imag.
- pré-condição: o número complexo é válido
- processo: pegar o valor real e imag. do n° complexo e colocar em um número real e um imaginário
- saída: 0 se tiver erro ou 1 se tiver sucesso
- pós-condição: números real e imag. com os valores do número complexo

d) libera - novo

- entrada: endereço do endereço de número complexo
- pré-condição: nenhuma
- processo: eliminar memória do n° complexo e limpar o endereço
- saída: nenhuma
- pós-condição: número complexo liberado

© Mattel

MONSTER HIGH

tilibra

Anotações / Anotaciones

1) soma

- entrada: endereço dos números complexos
- pré-condição: os dois números serem válidos
- processo: somar dois números
- saída: NULL se ocorrer erro e o endereço se der certo
- pós-condição: nenhuma

2) sub

- entrada: endereço dos números complexos
- pré-condição: entrada dos dois números complexos válidos
- processo: subtração dos dois números
- saída: o endereço do resultado ou NULL se der errado
- pós-condição: nenhuma

3) mult

- entrada: endereço dos números complexos
- pré-condição: ter os dois números complexos válidos
- processo: multiplicação dos números
- saída: endereço do resultado ou NULL se der errado
- pós-condição: nenhuma

4) create_ptc

- entrada: 3 valores double
- pré-condição: nenhuma
- processo: criar um ponto no R3 com valores inicializados
- saída: endereço de memória do ponto criado, ou NULL se tiver erro
- pós-condição: nenhuma

5) libera_ptc

- entrada: endereço do endereço de um ponto
- pré-condição: nenhuma
- processo: libera a memória do ponto criado e limpa o endereço

tilibra



- * saída: nenhuma
- * pós-condição: ponto liberado da memória



distância - pts

- * entrada: endereços dos dois pontos
- * pré-condição: ponto ser válido
- * processo: calcular a distância entre os pontos
- * saída: -1 se tiver erro e se der certo a distância dos pontos
- * pós-condição: nenhuma

03. cria - esfera

- * entrada: 6 valores para iniciar a esfera
- * pré-condição: nenhuma
- * processo: cria esfera com um ponto no centro e outro na superfície, com seus valores inicializados.
- * saída: endereço de memória da esfera criado ou null se tiver erro.
- * pós-condição: nenhuma

libera - esfera

- * entrada: endereço de endereço de uma esfera
- * pré-condição: nenhuma
- * processo: libera a memória da esfera e limpa o endereço
- * saída: nenhuma
- * pós-condição: esfera liberada da memória

© Mattel

Raio

- * entrada: endereço da esfera
- * pré-condição: esfera ser válida
- * processo: calcula o raio da esfera
- * saída: -1 se tiver erro ou o valor do raio
- * pós-condição: nenhuma





area

- * entrada = valor do raio da esfera
- * pré-condição = raio ser positivo
- * processo = calcular a área da esfera
- * saída = 1 se o raio for negativo ou o valor da área
- * pós-condição = nenhuma

volume

- * entrada = valor do raio da esfera
- * pré-condição = raio ser positivo
- * processo = calcular o volume da esfera
- * saída = 1 se o raio for positivo ou o valor do volume
- * pós-condição = nenhuma