

INF0286 – Algoritmos e Estruturas de Dados 1

Primeira prova – A01

Prof. Dr. Aldo André Díaz Salazar

Instruções

- A prova será realizada em equipe. Submeter apenas 1 prova por equipe em:

<https://classroom.github.com/a/W7Rqn1Ee>

- Todos os integrantes deverão participar da solução fazendo **commits**. Caso contrário a nota será de zero.
- **Prazo de entrega:** 30/11/2022, 23:59

Recomendações e descontos

- Utilizem comentários nos seus códigos. Será descontado 1 ponto “operação” ininteligível ou mal comentada.
- Utilizem indentação. Será descontado 1 ponto por “operação” mal indentada ou desorganizada.
- ★ O termo “operação” se refer a funções, procedimentos, trechos de código, arquivos de código fonte e instruções para a compilação do código.

(10.0p) Desafio – Números complexos

Projetem um TAD para representar números complexos e suas operações. Considerem os seguintes requisitos de design:

1. (2.0p) Os números complexos e suas operações serão representados na forma cartesiana $z = a + ib$. Não utilizem a representação polar $z = \rho e^{i\theta}$.
2. (3.0p) Utilizem uma abordagem **recursiva**, ao invés de iterativa, na implementação da operação de potenciação.
3. (2.0p) Implementem a função específica **COMPLEX0potencia2** para computar a potência do expoente complexo (vejam o exemplo com estrela “★” embaixo).
4. (0.5p) Utilizem 2 casas decimais e notação científica para a representação dos números. Por exemplo, o número $0.2 + 987i$ seria representado em notação científica como $2.00e-1 + 9.87e2i$.

5. (1.0p) Lembrem-se de documentar e organizar seu código fazendo uso de comentários e indentação.
6. (0.5p) Implementem um TAD **modular** incluindo um programa **main** por separado. Coloquem as instruções para a compilação do seu programa no arquivo **README.md** dentro do repositório da prova.
7. (1.0p) O código deverá compilar sem erros.
8. Na determinação de ângulos, utilizem a função trigonométrica **atan2**, ao invés de **atan**, para obter a redução no segundo quadrante.
9. Para facilitar o desenvolvimento, utilizem o modelo de implementação disponível na pasta **src** no repositório da prova. O modelo já implementa todos os resultados mostrados nos exemplos. Vocês somente precisam garantir que o programa “rode” corretamente.
10. Outras funções e variáveis auxiliares podem ser utilizadas na solução.

Exemplos

Dados os números complexos $z_1 = 3 + 4i$ e $z_2 = \sqrt{2} + \sqrt{2}i$, obtenha os seguintes resultados:

1. Adição: $z = 4.41 + 5.41i$
2. Conjugado de z_2 : $z_2^* = \sqrt{2} - \sqrt{2}i$
3. Multiplicação: $z_1 * z_2 = -1.41 + 9.9i$
4. Magnitude de z_2 : $|z_2| = 2$
5. Ângulo de z_2 : $\angle z_2 = \pi/4$ (radianos) ou 45 (graus)
6. Potência: $z_1^0 = 1$
7. Potência: $z_1^1 = 3 + 4i$
8. Potência: $z_1^5 = -2.37 \cdot 10^2 - 3.12 \cdot 10^3i$
9. Potência: $z_1^{-5} = -2.43 \cdot 10^{-5} + 3.19 \cdot 10^{-4}i$
- ★. Potência curiosa: $i^i = e^{-\pi/2} \approx 0.208$

Sugestões

- Caso precisem se lembrar dos números complexos e suas operações, o artigo da [Wikipedia](#) resulta ser bastante informativo.
- Para conferir o resultado numérico das suas operações, utilizem recursos online como o site da [WolframAlpha](#).
- Para referências sobre TADs, utilizem a aula **le04-tad.pdf**, slide 13, sobre a implementação de um TAD para um ponto 2D.
- Para referências sobre recursividade, utilizem a aula **le03-recursividade.pdf**, slide 8, sobre a implementação recursiva da operação de potenciação.

Saída do código

```

Numeros complexos a operar: z1 e z2
z = 3.00e+00 + 4.00e+00i
z = 1.41e+00 + 1.41e+00i

Adição: z1 + z2
z = 4.41e+00 + 5.41e+00i

Conjugado de z2: z2*
z = 1.41e+00 - 1.41e+00i

Multiplicação: z1 * z2
z = -1.41e+00 + 9.90e+00i

Magnitude de z2:
2.00

Ângulo de z2:
45.00 (graus)

Potência: z1 ^ 0
z = 1.00e+00 + 0.00e+00i

Potência: z1 ^ 1
z = 3.00e+00 + 4.00e+00i

Potência: z1 ^ 5
z = -2.37e+02 - 3.12e+03i

Potência: z1 ^ -5
z = -2.43e-05 + 3.19e-04i

Potência: i ^ i
z = 2.08e-01 + 0.00e+00i

```