

Universidade Federal do Piauí
Campus Senador Helvídio Nunes de Barros - Picos
Disciplina: Algoritmos e Programação II
Curso: Bacharelado em Sistemas de Informação
Professor: Glauber Dias Gonçalves

Lista de Atividades 05: ponteiros

Todas as questões abaixo devem ser implementadas na linguagem C.

1. Faça um programa para calcular o cubo de um número usando chamada de parâmetros por referência. Esse programa deve ter uma função seguindo o protótipo abaixo:

*void cubo(int *nPtr);*

Entrada:

A entrada consiste de vários inteiros, um valor por linha, correspondentes aos números para calcular o cubo. O final da entrada é marcado pelo valor (0).

Saída:

Para cada inteiro da entrada imprima o cubo correspondente, de acordo com o exemplo.

Entrada:

1

4

0

Saída:

1

64

2. Faça um programa que calcule as raízes de uma equação quadrática: $ax^2 + bx + c = 0$. Esse programa deve ter uma função com o seguinte protótipo:

*void raizes(float a, float b, float c, float *x1, float *x2);*

Essa função deve armazenar em x1 e x2 as raízes reais da equação, caso existam.

Entrada:

A entrada consiste de várias equações, três valores representando os coeficientes *a*, *b* e *c* por linha. O final da entrada é marcado pelo valor (0).

Saída:

Para cada equação da entrada imprima os valores x1 e x2 correspondente, e o nome “complexo” para indicar que as raízes não são números reais, de acordo com o exemplo.

Entrada:

4.0 4.0 1.0

1.0 -5.0 6.0

1.0 -4.0 5.0

0

Saída:

-0.5 -0.5

3.0 2.0

complexo

3. Faça um programa que converta um vetor de caracteres para maiúsculo. Esse programa deve ter uma função com o seguinte protótipo:

`void maiusculo(char *sPtr);`

Dica: use `*sPtr` para dereferenciar cada caractere do vetor e `++sPtr` para avançar para o próximo caractere.

Entrada:

A entrada consiste de várias linhas, um vetor de caractere por linha. O final da entrada é marcado pelo valor (0).

Saída:

Para cada vetor imprima o vetor correspondente em maiúsculo, de acordo com o exemplo.

Entrada:

characters and \$32.98

hoje eh dia de estudar c++

0

Saída:

CHARACTERS AND \$32.98

HOJE EH DIA DE ESTUDAR C++

4 - Um número complexo é escrito na forma algébrica $a + bi$, onde a e b são números reais, o valor de a é a parte real do número complexo e o valor bi é a parte imaginária. A soma de dois números complexos é realizada somando as partes reais separadamente das partes imaginárias.

Por exemplo, seja os números $z1 = 6 + 5i$ e $z2 = 2 - i$, a soma $z1 + z2 = z3$, onde $z3 = 8 + 4i$.

Escreva um programa para somar dois números complexos. Esse programa deve ter uma função com o seguinte protótipo:

`void somaComplexo(float rz1, float iz1, float rz2, float iz2, float *rz3, float *iz3);`

Note que as variáveis r representam a parte real e as variáveis i representam a parte imaginária dos numeros complexos $z1$, $z2$ e $z3$.

Entrada:

A entrada consiste de vários pares de números complexos, em cada linha os dois primeiros valores representam as partes real e imaginária do primeiro número e os dois números seguintes, as partes real e imaginária do segundo número. O final da entrada é marcado pelo valor (0).

Saída:

Para cada par de números complexos da entrada, imprima os números resultantes, de acordo com o exemplo.

Entrada:

6.0 5.0 2.0 -1.0

7.0 3.0 9.0 3.0

0

Saída:

8.0 + 4.0i

16.0 + 6.0i

5. Faça um programa que simule o embaralhamento de cartas, assim como a questão 3 da atividade04. Esse programa deve usar a estrutura *Carta* com os campos *face* e *naipe* ter funções seguindo os protótipos:

`void preenche(Carta *deck, char *face[], char *naipe[]);
void embaralha(Carta *deck);
void imprime(Carta *deck);`

Dica:

- use o vetor de *strings* “*face*” para armazenar os nomes das treze faces de cartas
- use o vetor de *strings* “*naipe*” para armazenar os nomes dos quatro naipes do baralho
- use subscrito nos vetores, por exemplo `deck[i]`, `face[i]` e `naipe[i]`, para iterar sobre vetores de *Carta* e *strings* recebidos como ponteiros.