

Listas de Exercícios
Disciplina de Fundamentos de Programacao

Prof. David Sena Oliveira
Universidade Federal do Ceara - Quixadá

Outubro de 2013

Instruções

Cada questão possui:

- instruções de implementação,
- um protótipo da função a ser implementada e
- uma função de teste.

Seja o seguinte exemplo:

Questão Exemplo

Objetivo: Somar dois números inteiros.

Entrada : Dois números inteiros.

Saída : A soma dos números.

```
int soma (int num1, num2);
void test()
{
    puts(soma(6, 4) == 10);
    puts(soma(3,-4) == -1);
}
```

Você deve implementar a função soma e executar o código de teste. O ambiente completo do arquivo main.c seria algo parecido com o exemplo a seguir.

```
#include <stdio.h>

int soma (int num1, num2)
{
    return num1 + num2; //essa aqui é a sua implementacao
}

void test()
{
    puts(soma(6, 4) == 10);
    puts(soma(3,-4) == -1);
}

int main()
{
    test();//aqui voce chama o codigo de teste.
    return 1;
}
```

Se sua implementação da função estiver correta, como aconteceu na função soma, será impresso o número 1 na tela. Caso contrário, será impresso o número 0. Assim, fica mais fácil de acompanhar em quais casos seu código está falhando.

Módulos de Tipos e Operações

Aluno Autor:

Natasha da Silva Sauro

Essa lista aborda a parte básica sobre operações e tipos. Boa parte dos exercícios visa alertar sobre peculiaridades da linguagem C, bem como operações entre tipos. Também visa acostumar o aluno sobre o uso de funções para resolver problemas.

São cobertos os tipos: int, char, float.

E as operações básicas: +, -, /, * e módulo, && e ||.

1 Fáceis

1.1 Pares ou ímpares

Objetivo: Utilizar a operação de módulo (%) para verificar se um número é par ou ímpar.

Entrada : Um número inteiro.

Saída : 1, se o número for par.
0, se o número for ímpar.

```
int eh_par (int num);  
void test()  
{  
    puts(eh_par(6) == 1);  
    puts(eh_par(3) == 0);  
}
```

1.2 Cast pra float

Objetivo: Calcular uma operação de divisão entre inteiros utilizando cast para float.

Entrada: Dois números inteiros.

Saída : O resultado da divisão em float.

```
float divide (int n1, int n2);  
void test()  
{  
    puts(divide(6 , 4) == 1.5);  
    puts(divide(4.5, -3) == -1.5);  
}
```

1.3 Operações básicas

Objetivo: Realização de operações básicas entre floats

Entrada : Dois números a serem operados

Saída : O resultado da operação

```
float soma ( float num1, float num2 );  
float subtrai( float num1, float num2 );  
float multi ( float num1, float num2 );  
float divide ( float num1, float num2 );  
  
void test()  
{  
    puts(soma (4.5, 5.5) == 10);  
    puts(subtrai(3.1, 2.0) == 1.1);  
    puts(multi (3.2, 2 ) == 6.4);  
    puts(divide (5 , 2 ) == 2.5);  
}
```

1.4 Operadores lógicos

Objetivo: Utilizar operadores lógicos para verificar se um número é divisível por 2 e 3, mas não pelo número 5.

Entrada : Um número inteiro.

Saída : 1, se for divisível por 2 e por 3, mas não por 5.
0, caso contrário

```
float div_23n5 (int num);  
void test()  
{  
    puts(div_23n5(2) == 0);  
    puts(div_23n5(3) == 0);  
    puts(div_23n5(6) == 1);  
    puts(div_23n5(12) == 1);  
    puts(div_23n5(30) == 0);  
}
```

2 Questões Médias

2.1 Conversão de tempo

Objetivo: Calcular o número de segundos que passaram desde a meia noite até a hora informada.

Entrada : Horas, minutos e segundos.

Saída : O tempo de segundos

```
int meia_noite( int hora, int minuto, int segundo );

//melhorar esse teste
void teste_meia_noite()
{
    puts(meia_noite(0, 0, 1) == 1);
    puts(meia_noite(0, 1, 1) == 61);
}
```

2.2 Média Ponderada

Objetivo: Calcular a média ponderada entre duas notas

Entrada : As duas notas e seus respectivos pesos

Saída : Retornar o valor da média como float

```
float media_pond( float nota1, float nota2, float peso1, float peso2 );

void test_media_pond()
{
    puts(media_pond(8, 8, 1, 1) == 8);
    puts(media_pond(8, 6, 2, 1) == 7);
    puts(media_pond(0,10, 9, 1) == 1);
    puts(media_pond(8, 4, 1, 3) == 5);
}
```

3 Questões Divertidas

3.1 Ano bissesto

Objetivo: Implementar a função que calcula se um ano é bissesto ou não.

Entrada : 0 ano.

Saída : 1 se for bissesto.
0 se não for.

```
int bissesto( int ano );  
void test()  
{  
    //codigo de teste aqui  
}
```

3.2 Dia da Semana

Objetivo: Implementar a função que retorna o dia da semana.

Entrada : 0 ano, o mês e o dia do mês.

Saída : 1 para domingo, 2 para segunda, 3 para terça, ...

```
int dds( int ano, int mes, int dia );  
void test()  
{  
    //codigo de teste aqui  
}
```