

SME0230 - Introdução à Programação de Computadores

Professora: Marina Andretta (andretta at icmc.usp.br)

Estagiário PAE: Pâmela Cândida (pamela at icmc.usp.br)

Monitor: Matheus Doretto Compri (matheusdc at grad.icmc.usp.br)

Forma de entrega: O exercício desta aula prática deverá ser entregue por e-mail para

- andretta at icmc.usp.br
- pamela at icmc.usp.br
- matheusdc at grad.icmc.usp.br

Ele deverá estar no formato .c e o nome do arquivo deverá ser numeroUsp_i.c. Exemplo: 6711776_1.c, onde *i* representa o número do exercício.

O título da mensagem será: IPC2011_aula10

Observações importantes:

1. não serão recebidos trabalhos após dia 20/05 23h59;
2. os dois códigos (.c) devem ser entregues no mesmo mail com os nomes: numeroUsp_1.c e numeroUsp_2.c;
3. trabalho entregue com nome diferente do padrão acima citado NÃO será corrigido, i.e, nota = 0;
4. o aluno que enviar o trabalho fora do padrão ou que não enviar para os e-mails especificados receberá zero como nota do referente exercício da aula prática.
5. **Cuidado com saída!!!!!! Faça exatamente como pede o enunciado! Observe os exemplos de saída!**

Observação/ Pedido: envie os códigos (.c) anteriores novamente. Caso tenha dúvidas, veja o pdf das aulas, estão atualizados. Os códigos a serem re-enviados são:

1. Aula de laboratório do dia 26/04: envie e-mail com o título IPC2011_aula7, os códigos são: 1) média ponderada, SEM estrutura de repetição, nomeado como numeroUsp_1.c; 2) dizer se triângulo é isosceles, equilatero ou escaleno, nomeado como numeroUsp_2.c
2. Aula de laboratório do dia 03/05: envie e-mail com o título IPC2011_aula8, os códigos são: 1) média ponderada, COM estrutura de repetição, nomeado como numeroUsp_1.c; 2) calculadora, SEM estrutura de repetição, nomeado como numeroUsp_2.c
3. Aula de laboratório do dia 10/05: envie e-mail com o título IPC2011_aula9, os códigos são: 1) calculadora, COM estrutura de repetição, nomeado como numeroUsp_1.c; 2) aproximação da raiz quadrada, nomeado como numeroUsp_2.c

Utilize esta chance para verificar seu código, padrão de saída, principalmente! Nunca coloque um printf("Digite bla bla bla...") a menos que esteja absolutamente descrito no enunciado! Você pode ficar com zero por falar demais!

Aula Prática 10 - Introdução à programação

Durante a aula de laboratório... exercício 1

Você deve ler um valor, validá-lo e, caso não válido, dar outra chance ao usuário. Qual o melhor comando para se implementar isso? Usando a melhor prática de programação, sorteie um número entre 0 a 100. Depois dê 3 chances para o usuário adivinhar este número. A resposta do programa deve ser “você acertou na tentativa *i*” ou “azar no jogo, sorte no amor”. O usuário pode continuar jogando quantas vezes quiser, para isso ele responde ('S' ou 's') para a pergunta “deseja continuar?”. Para encerrar o programa, responde 'N' ou 'n'. Caso entre com um valor diferente para a pergunta “deseja continuar?”, deve-se dizer “resposta inválida! para continuar S ou s e para encerrar N ou n”.

Exemplo de Entrada

2
3
7
1

Exemplo de Saída

(supondo que o número sorteado foi 7)

```
 você acertou na tentativa 3
deseja continuar?
```

(ou supondo que o número sorteado foi 99)

```
 azar no jogo, sorte no amor
deseja continuar?
```

***OBS:** este trabalho não será corrigido automaticamente, pois depende do valor aleatório. Vamos analisar se além de correto, o programa atende a bons padrões de programação: indentação, modularidade, lógica, melhor escolha dos comandos de condição e repetição (if, while, for, etc) de forma a ter um código compacto*

Para fazer em casa... exercício 2

O usuário deseja saber se um conjunto de números são primos ou não. Para isso ele digita esses números (inteiros entre 2 e 1000) separados por algum caracter de tabulação (`\t`, `\n`, `' '`). O programa deve separar esses números em dois conjuntos para dizer quais são primos e quais não são. Importante a saída fornecer os números na mesma ordem de entrada, dizendo primeiro os primos e, na próxima linha, os não primos. Se o número for maior que 1000 ou menor que 2, desconsidere-o.

Exemplo de Entrada

```
10
2
6
7
3
1000
```

Exemplo de Saída

primos: 2, 7, 3
não primos: 10, 6, 1000

Exemplo de Entrada

1001
2
3
4
5
6
-1
22222
7
8

Exemplo de Saída

primos: 2, 3, 5, 7
não primos: 4, 6, 8

***OBS:** lembre que para o scanf não faz diferença o separador entre os elementos da entrada!!! A entrada é formatada! Isso significa que o scanf vai procurar por um número, descartando todos os caracteres de tabulação que encontrar no “meio do caminho”, ok?*

***OBS 2:** Há um algoritmo famoso para saber se um número é ou não primo. Google it!*