Algoritmos e Estruturas de Dados I (DCC/003)

Aulas Práticas – Parte 1 do curso
Data de entrega: 09-05-2017

Orientações gerais:

- Os exercícios deverão ser preferencialmente feitos durante o tempo das aulas de laboratório;
- As dúvidas serão esclarecidas nas aulas;
- Os exercícios deverão ser entregues via Moodle;
- Os executáveis não devem ser enviados, apenas os arquivos de extensão ".c".

Primeiros programas e Estruturas básicas

Orientações:

a) O objetivo dos primeiros exercícios é a familiarização com o ambiente de programação e um primeiro contato com a linguagem C e algumas de suas características, como o fato de todos as linhas terminarem com ponto-e-vírgula.

b) Algumas das questões abaixo já incluem códigos. Você pode usar estes códigos como ponto de partida para as demais questões, alterando-os como necessário.

c) Usaremos, nos exercícios abaixo, o conceito de variáveis. Variáveis em C se comportam de forma distinta de variáveis da matemática mas, inicialmente, podemos supor que são similares. Para usar uma variável, precisamos declará-la, ou seja, informar o computador de que usaremos tal variável. Para declarar uma variável, basta dizer seu tipo e o nome escolhido. Alguns exemplos:

- `float x;`
  - Neste caso declaramos uma variável de nome "x", do tipo float;
- `double y;`
  - Similar ao caso anterior, declaramos uma variável de nome "y", do tipo double;
- `float a, b, c;`
  - Desta vez, estamos usando uma única linha para declarar três variáveis do tipo float, chamadas "a", "b" e "c".

d) Algumas operações matemáticas úteis são:

- Soma: `a + b`
- Subtração: `a - b`
- Multiplicação: `a * b`
- Divisão: `a / b`
- Raiz quadrada: `sqrt(a)`
- Seno: `sin(a)`

Para usar algumas funções matemáticas, pode ser necessário o uso da seguinte linha no cabeçalho de seu programa:

```
#include <math.h>
```

e) Para atribuir um valor a uma variável, usamos o operador de atribuição "=". Neste caso, a variável onde o valor será atribuído deve estar à esquerda do símbolo "=". Exemplos:

- `x = 10;`
  - Guarda o valor 10 na variável x;
- `y = x;`
  - Neste caso o valor armazenado na variável x, qualquer que seja, será copiado para a variável y;
- `z = x + y*y;`

- Neste caso, o valor de z será o valor de x somado ao quadrado de y. Caso os três comandos acima sejam executados nesta ordem, então o valor de z será 110.

f) A função `printf()` é usada para a impressão de dados na tela. Por exemplo, `printf("Oi")` imprimirá a mensagem "Oi" na tela.

- Em determinados casos, para imprimir números, precisamos fornecer informações adicionais à função printf. Para cara símbolo % (porcentagem) o comando printf lerá o conteúdo de uma variável e o escreverá na tela.
- Nesta aula, ao imprimir um dado do tipo "float" usaremos `printf("%.2f", x)`, para imprimir o valor de x.
- Para dados do tipo "double" usaremos `printf("%lf", x)`.

1) Escreva um código para calcular o seno de um valor informado (como 30), conforme apresentado a seguir. Compile o código, verificando se algum erro ocorreu. Execute o código com diversos valores de ângulos e verifique se o resultado apresentado está correto.

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define PI 3.14159265

/*
Comentários... 1o programa em C
*/

int main(int argc, char *argv[])
{
    double graus, resultado;
    graus = 30.0;
    resultado = sin (graus * PI / 180);
    printf("O valor do seno = %lf", resultado);
    printf("\n");
    return 0;
}
```

[salve o seu código com o nome: **ap01ex1.c**]

2) Uma conta poupança foi aberta com um depósito de R$500,00. Esta conta é remunerada em 1% de juros ao mês. Qual será o valor da conta após três meses?

As variáveis **p**, **s** e **t** são mesmo necessárias no programa abaixo, se quisermos apenas saber o valor existente na conta após passados os 3 meses? É possível refazer o programa usando apenas a variável **d**?

```c
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
  float d,p,s,t;

  d = 500;   // depósito inicial
  // após o primeiro mês
  p = d + 0.01*d;
  // após o segundo mês
  s = p + 0.01*p;
  // após o terceiro mês
  t = s + 0.01*s;
  printf("Valor da conta = %.2f\n",t);
  system("pause");
  return 0;
}
```

[salve o seu código com o nome: **ap01ex2.c**]

3) Elaborar um programa em Linguagem C para resolver o seguinte problema:

Considere que os valores (inteiros e positivos) para as variáveis **a**, **b** e **c** correspondem aos lados de um triângulo retângulo com catetos **a** e **b**, e hipotenusa **c**. Determinar a área do triângulo pela fórmula:

$área = \sqrt{s*(s-a)*(s-b)*(s-c)}$, onde $s = \frac{a+b+c}{2}$

[salve o seu código com o nome: **ap01ex3.c**]

4) Escreva um código para calcular e imprimir alguns valores importantes sobre os valores que podem ser armazenados em uma variável do tipo inteiro (int x).
Use a função "pow" (potência), sabendo que:
- `pow(x,y)` calcula x^y;
- um inteiro possui 32 bits assim o maior valor inteiro positivo será 2^{31}-1;

a) Usando a função potência, calcule este maior valor para o tipo inteiro e imprima em seguida.
b) Em seguida, você deverá somar 1 ao valor e novamente imprimir para ver o que acontece.

Use %d (decimal) para imprimir o valor do número.

[salve o seu código com o nome: **ap01ex4.c**]

5) Elabore um programa em Linguagem C para calcular as raízes da equação $ax^2 + bx + c = 0$. Para isso você precisa apenas das variáveis a, b, c, mas não da variável x. Teste seu programa com os valores a = 2, b = 10 e c = 12. A resposta está correta?

Dica: uma das raízes pode ser calculada como
```
r1 = (-b + sqrt(b*b - 4*a*c))/(2*a);
```
Todos os parênteses são necessários?

[salve o seu código com o nome: **ap01ex5.c**]

Estruturas de controle

1) Iremos fazer um programa para imprimir na tela o rendimento semestral global (RSG) de um aluno na UFMG. O programa a ser desenvolvido irá receber as notas de quatro disciplinas (nota1, nota2, nota3 e nota4), notas estas de 0 a 100, bem como a quantidade de créditos de cada disciplina (cred1, cred2, cred3 e cred4). Os créditos poderão ser valores de 20 a 60. Vamos assumir que o aluno não trancou nenhuma disciplina.

O seu programa deve imprimir mensagens de erro se os valores forem inválidos (notas fora da faixa 0-100 e créditos fora da faixa 20-60, ambos sendo intervalos fechados).
No cálculo do RSG na UFMG, convertem-se os conceitos obtidos em cada atividade/ disciplina em valores, observando-se a seguinte correspondência:

Conceito	Valor
A	5
B	4
C	3
D	2
E	1
F	0

O valor do conceito de cada atividade em que o aluno se matriculou no semestre, excluídas as porventura trancadas, é multiplicado por seu respectivo número de créditos; os produtos assim obtidos são somados e o resultado é dividido pelo número total de créditos em que o aluno se matriculou no semestre.

Execute o código para que o resultado seja apresentado.
[salve o seu código com o nome: **ap02ex1.c**]

2) O cálculo do IMC é feito a partir da divisão do peso pela altura ao quadrado. Por exemplo, uma pessoa que pesa 80kg e que tem altura de 1,80 m, terá um IMC de $80/ (1.80)^2 = 24,69$.
Faça um programa que leia o peso e a altura da pessoa, e imprima na tela o valor do IMC, bem como indique a situação da pessoa, baseado no seu IMC.

Resultado	Situação
Abaixo de 17	Muito abaixo do *peso*
Entre 17 e 18,49	Abaixo do *peso*
Entre 18,5 e 24,99	*Peso* normal
Entre 25 e 29,99	Acima do *peso*
Entre 30 e 34,99	*Obesidade* I
Entre 35 e 39,99	*Obesidade* II (severa)
Acima de 40	*Obesidade* III (mórbida)

Compile, execute e teste o seu código implementado. [código: **ap02ex2.c**].

3) Considere a seguinte definição de ano bissexto (ano em que o mês de fevereiro tem 29 dias):

- Um ano não divisível por 100 e divisível por 4 é bissexto;
- Um ano divisível por 400 é bissexto;
- Os demais anos **não** são bissextos.
    - Lógica a ser implementada:
        - (não divisível por 100 E divisível por 4) OU (divisível por 400);

Elabore o seu código principal do programa, que deve fazer o seguinte:

- Recebe do usuário (como valor de entrada) o ano;
    - Receba um valor inteiro que é o ano a ser testado;
- Teste SE (if) o ano é bissexto ou NÃO (else);
- Imprime na tela para o usuário a mensagem “O ano <valor do ano> informado é bissexto” ou então “O ano <valor do ano> informado não é bissexto”.

Execute o código para testar seu programa de computador e verificar se o resultado está correto. [salve o seu código com o nome: **ap02ex3.c**]

Estruturas de repetição e Funções

1) Escreva um programa em C para cálculo do máximo divisor comum (MDC) entre três números.

Dica: O mdc(a,b,c) = mdc(a, mdc(b,c))

Exemplo: caso a entrada seja 156, 182, 429, sua saída deve ser 13.

Execute o código para testar seu programa de computador e verificar se o resultado está correto. [salve o seu código com o nome: **ap03ex1.c**]

2) Escreva um programa que leia um número inteiro N e diga se ele é um número primo ou não. Um número é primo se ele é pelo menos 2 e só é divisível por 1 e por ele mesmo.

Exemplo 1: caso a entrada seja 5, a saída deve ser:
SIM

Exemplo 2: caso a entrada seja 323, a saída deve ser:
NAO

Dica: um número A é divisível por B se o resto da divisão de A por B é zero.

Execute o código para testar seu programa de computador e verificar se o resultado está correto. [salve o seu código com o nome: **ap03ex2.c**]

3) Escreva um programa que leia dois inteiros N e M, e imprima todos os números ímpares maiores que N e menores que M.

Exemplo: caso N = -4 e M = 7, a saída deve ser:
-3 -1 1 3 5

Execute o código para testar seu programa de computador e verificar se o resultado está correto. [salve o seu código com o nome: **ap03ex3.c**]

4) Escreva um programa que leia um número positivo N e imprima N linhas do triângulo ordenado. Um triângulo ordenado pode ser visto abaixo, para o caso N = 6.

1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21

Note que foram impressas exatamente N = 6 linhas e a i-ésima linha contém exatamente i números.

Execute o código para testar seu programa de computador e verificar se o resultado está correto. [salve o seu código com o nome: **ap03-ex4.c**]

5) Uma empreiteira paga seus pedreiros por metros quadrados de serviços produzidos **diariamente**, adicionando também o valor do almoço e passagem para irem e voltar do trabalho. Essa construtora procurou você para desenvolver um sistema que calcula quanto ela deve pagar para cada um de seus pedreiros. Apresente um programa que resolva o problema citado.
- O programa deverá receber:
    - O valor do vale-alimentação;
    - O valor do transporte unitário (sendo que cada dia o pedreiro usa 2 vales);
    - Quantos metros quadrados trabalhou;
- O valor do metro quadrado é calculado da seguinte forma:

Quantos metros quadrados trabalhou?	**Valor do metro quadrado**
Menos de 10m2	R$ 10,00
Mais ou igual a 10m2 até 20m2	R$ 11,50
Mais ou igual a 20m2	R$ 13,00

- Crie a lógica para calcular o valor a ser pago ao pedreiro;
- Depois na função principal, receba os valores especificados, execute os cálculos e armazene o valor do pagamento;

- Ao final, o programa deve imprimir a seguinte mensagem ao final de sua execução: “O pedreiro deve receber R$ x.”, onde x é o valor a ser pago. [salve o seu código com o nome: **ap03-ex5.c**]

6) Escreva as seguintes funções utilizando, se necessário, estruturas condicionais e de repetição em C. Teste sua solução no computador. [salve o seu código com o nome: **ap03-ex6.c**]

a) fat(n): retorna o valor do fatorial de n.

Exemplo: fat(5) deve retornar 120.

b) mdc(a,b): retorna o máximo divisor comum entre a e b.

c) mdc4(a,b,c,d): retorna o máximo divisor comum entre a, b, c e d.

d) multiplos(n, x): procedimento que escreve na tela todos os múltiplos de n entre 0 e x, inclusive, separados por vírgula.

Exemplo: multiplos (3, 21) deve imprimir:

0, 3, 6, 9, 12, 15, 18, 21

Note que não deve aparecer uma vírgula após o último número e que 0 sempre será impresso.

e) primo(x): retorna 1 se x é primo e 0 em caso contrário.

f) decrescente(x): procedimento que escreve uma seqüência de inteiros menores que x e maiores que 0.

Exemplo: decrescente(6) deve imprimir:

5 4 3 2 1

g) pa(a, r, n): retorna a soma dos termos de uma Progressão Aritmética de **n** termos, com termo inicial **a** razão **r**.

h) dig(n): retorna a soma dos dígitos de um inteiro positivo n.

Exemplo: A soma dos dígitos de 132, por exemplo, é 6. Já a soma dos dígitos de 1095 é 16.

i) mmc(a, b): retorna o menor múltiplo comum de a e b.

j) n_esimo_primo(n): retorna o n-ésimo primo.

Exemplos:

`n_esimo_primo(1)` deve retornar 2;
`n_esimo_primo(2)` deve retornar 3;
`n_esimo_primo(17)` deve retornar 59;

k) hora (h, m, s): recebe três inteiros **h**, **m** e **s**, correspondentes a hora, minuto e segundo e imprime o horário um segundo depois.

Exemplo: hora (16, 06, 59) deve imprimir 16:07:00
hora (23, 59, 59) deve imprimir 00:00:00

7) Escreva as funções abaixo, que desenham as figuras indicadas. [salve o seu código com o nome: **ap03-ex7.c**]

a) triangulo(n): procedimento que recebe um inteiro n e “desenha” na tela o triângulo de base e altura n (veja exemplo abaixo para n = 10).

```
**********
*********
********
*******
******
*****
****
***
**
*
```

b) balao(n): procedimento que recebe um inteiro n e “desenha” na tela o balão de altura 2*n (veja exemplo abaixo para n = 6).

```
************
*****  *****
****    ****
***      ***
**        **
*          *
*          *
**        **
***      ***
****    ****
*****  *****
************
```

c) arte(n): procedimento que recebe um inteiro n e “desenha” na tela a arte de altura 2*n (veja exemplo abaixo para n = 5).

```
\     *     /
 \   ***   /
  \*****/
   \***/
    \*/
    /*\
   /***\
  /*****\
 /   ***   \
/     *     \
```

d) tabuleiro(n): procedimento que recebe um inteiro n e “desenha” na tela um tabuleiro de xadrez de altura e largura 2*n (veja exemplo abaixo para n = 4).

```
.#.#.#.#
#.#.#.#.
.#.#.#.#
#.#.#.#.
.#.#.#.#
#.#.#.#.
.#.#.#.#
#.#.#.#.
```

Matrizes e vetores

1) Escreva um programa em C para praticar o conceito de variável indexada (vetor) agora usando valores reais.

Você deverá primeiro declarar um vetor de 7 posições do tipo real (float vetorFloat[7]).

Depois vai criar duas estruturas de repetição usando comando FOR, a saber:
- A primeira vai ler os valores digitados do usuários e ir armazenando no vetor;
- A segunda estrutura irá imprimir os valores informados pelo usuário na tela;

Por fim, faça uma outra estrutura de repetição que imprima apenas os valores pares do índice do vetor, ou seja, as posições 2, 4 e 6.

Execute o código para testar seu programa de computador e verificar se o resultado está correto. [salve o seu código com o nome: **ap04-ex1.c**]

2) Escreva um programa em C para receber N valores informados pelo usuário, onde N será informado no início da execução pelo usuário.

Defina também uma constante MAX, que é o valor máximo de posições do vetor, que será sempre o limite máximo que alocou para vetor, tendo N um valor menor.
- #define MAX 100

Depois vai criar duas estruturas de repetição usando comando FOR, a saber:
- A primeira vai ler os valores digitados do usuários e ir armazenando no vetor;
- A segunda estrutura irá imprimir os valores do vetor em ORDEM INVERSA.

Execute o código para testar seu programa de computador e verificar se o resultado está correto.
[salve o seu código com o nome: ap04-ex2.c]

3) Defina estaticamente as dimensões máximas das matrizes:
- #define MAX-Linhas 12
- #define MAX-Colunas 12

Faça um programa que vai realizar a SOMA de 2 matrizes e atribuir o resultado a uma MATRIZ RESULTADO.

Um usuário vai definir via entrada de dados as dimensões das matrizes LIN e COL (que devem ser menores que os limites estabelecidos).

- RESULTADO = A + B
  - A e B devem ter as mesmas dimensões
  - Cada elemento de C é igual à soma dos elementos correspondentes de A e B:

- $C_{ij} = A_{ij} + B_{ij}$

Execute o código para testar seu programa de computador e verificar se o resultado está correto. [salve o seu código com o nome: **ap04-ex3.c**]

Desafio: Exercício opcional – Para pensar e treinar a lógica e programação

Dois números inteiros são ditos amigos se a soma dos divisores de cada um deles (menores que eles) é igual ao outro. [salve o seu código com o nome: **ap04-desafio.c**]

Por exemplo:
- Os divisores de **220** são 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 e 110 e
1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = **284**

e os divisores de **284** são: 1, 2, 4, 71 e 142 e
1 + 2 + 4 + 71 + 142 = **220**.

Escreva um programa que determine todos os pares de inteiros amigos menores que