UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL
IMD0012.1 - PRÁTICAS DE TÉCNICAS DE PROGRAMAÇÃO - T03 (2017.1)

Avaliação I
Unidade I - 28.03.2017

Aluno:

Instruções Gerais

Leia atentamente as instruções antes de iniciar a avaliação:

1) Não esqueça de colocar o seu nome completo nesta folha.
2) Esta é uma prova **individual** e **sem consulta**;
3) Você terá duas aulas (totalizando 100 minutos) para responder a todas as questões. Por isso, **gerencie bem o seu tempo**;
4) A prova vale 10,0 pontos no total. O valor de cada questão é fornecido junto ao enunciado;
5) Caso você tenha resolvido corretamente 90% (ou mais) dos exercícios propostos no URI (no total de 98 exercícios) e entregue a Tarefa do Campeonato de Futebol (arranjos multidimensionais) corretamente implementado e funcionado, você pode indicar UMA das questões marcada como "Promoção" para ser substituída por estas conquistas, sendo o total de pontos da questão atribuída em sua totalidade. Lembre-se que o código do exercício do campeonato deverá ser entregue via SIGAA. Já a verificação do URI será via sistema;
6) Leia atentamente o enunciado de cada questão antes de iniciar a sua resposta;
7) As implementações devem ser feitas em linguagem C e os arquivos de sua implementação deverão ser enviados via SIGAA na atividade Avaliação 01;

   ** Crie um arquivo compactado, em formato ZIP, com a nomenclatura Avaliacao01.ZIP. Os códigos-fontes deverão ser salvos nesta pasta, com o nome do arquivo indicando a questão (Ex.: 01.c, 02.c, 02A.c, 02B.c, etc). Certifique-se de que os arquivos foram salvos corretamente antes de enviá-los através do SIGAA, na tarefa específica para esta avaliação.
8) Esta folha de questões, devidamente identificada, deve ser entregue ao professor ao final da avaliação;
9) Decréscimos de nota poderão ser aplicados caso sejam observadas as seguintes faltas:

Falta	Decréscimo
Programa apresenta erros de compilação, não executa ou apresenta saída incorreta	–50%
Legibilidade comprometida (falta de indentação, etc)	–20%
Programa compila com mensagens de aviso (*warnings*)	–30%
Plagiarismo (no caso de cópia de colega, ambos serão penalizados)	–100%

Questão 01 (3,00). Dizemos que um inteiro positivo **N** é perfeito se for igual à soma de seus divisores positivos diferentes de **N**. Por exemplo, 6 é perfeito, pois 1+2+3 = 6.

Implemente um programa, onde dado um inteiro positivo **M**, sejam impressos todos os números perfeitos menores que **M** (ou seja, entre 0 e **M**).

Um exemplo de execução do seu programa, poderia ser:

```
$ ./questao01
Entre com o valor de M: 500
Numeros perfeitos entre 0 e 500: 6 28 496
```

Questão 02 (2,00). 14. Implemente um programa que leia **N** valores reais (até que o valor -999 seja digitado), armazene-os em um vetor e o compacte, ou seja, elimine as posições com valor zero avançando uma posição, com os com os valores subsequentes do vetor. Dessa forma todos os "zeros" devem ficar para as posições finais do vetor.

Um exemplo de execução do seu programa, poderia ser:

```
$ ./questao02
Entre com os valores: 0 2.0 5 -66.3 0.0 17 0 0.0 -8.5 90.7 0 4.1 0 -999

Vetor compactado: 2.0 5 -66.3 17 -8.5 90.7 4.1 0 0.0 0 0.0 0 0
```

Questão 03 (2,00). Príncipe Adam foi desafiado por Mentor a criar um programa em linguagem C que verifique se um dado número é ou não palíndromo. Um número inteiro positivo, com mais de um dígito, é chamado palíndromo, se os seus dígitos da direita para a esquerda ou da esquerda para a direita, tenham o mesmo valor. Por exemplo, o número 32423 é palíndromo. Assim como os números 44, 212 e 12355321.

Segundo as instruções de Mentor, o programa deverá ler um número inteiro positivo X e imprimir como saída "O numero X e um numero palindromo.", caso o número seja palíndromo; OU "O numero X não e um numero palindromo.", caso contrário.

Após muito pensar, o príncipe Adam apresentou o código fonte abaixo à sua amiga Tila, antes de entrega-lo ao Mentor. Como era de se esperar, a filha de Mentor já sabia programar bem em C e disse ao príncipe que ele estava "Quase lá!", mas que haviam alguns detalhes a corrigir.

```
/* Programa que verifica se um numero é palindromo */

#include <stdio.h>

int main() {
  int num,
  aux,
  reverso;

  printf("Digite um inteiro positivo: ");
  scanf("%d", num);

  aux = num;
  reverso = 0;

  while (aux != 0) {
    reverso = reverso * 100 + aux % 10;
    aux = aux / 10;
  }

  if (reverso = num)
    printf("%d nao e um numero palindromo \n", num);
  else
    printf("%d e um numero palindromo \n", num);

  return 0;
}
```

Avalie o código acima, apresentado pelo príncipe Adam, e aponte os detalhes que devem ser corrigidos (provavelmente os mesmos detalhes apontados por Tila). Vá além e indique ainda como corrigir cada detalhe, sem alterar a lógica de programação do príncipe Adam.

Questão 04 (3,00). Simule a execução do programa mistério abaixo destacando a sua saída para o conjunto de dados dado a seguir. Não é para implementar, mas sim para ler o código, entender e simular o seu funcionamento! A seguir, descreva, de forma sucinta e objetiva, o que o programa faz.

Para ajudar a compreender o funcionamento do programa, teste-o para o seguinte conjunto de pares de entrada.

```
    2    3
    5    2
    7    1
    0    5
    3    2
```

```
/* Programa misterio */

#include <stdio.h>

int main()
{
  int a, b, x, y, t, i;

  printf("Digite um par de numeros: ");
  scanf("%d %d", &a, &b);
  printf("(%d, %d)\n", a, b);
  x = 0;
  y  = 0;
  while (a != 0) {
    x = x + 1;
    t = 1;
    for (i = 1; i <= b; i++)
      t = t * a;
    printf("ValorX = %d\n", t);
    y = y + t;
    printf("ValorY = %d\n", y);
    printf("Digite um par de numeros: ");
    scanf("%d %d", &a, &b);
    printf("(%d, %d)\n", a, b);
  }
  printf("Total de pares: %d\n", x);
  return 0;
}
```

Fim da Avaliação.