UNIVERSIDADE FUMEC/FACE
Faculdade de Ciências Empresariais
Curso: Ciência da Computação – BH
Disciplina: ED-I – Professor: Ricardo Terra

INFORMAÇÕES SOBRE ESTA ATIVIDADE:

- Esta atividade valerá 2,0 pontos (*1,0 ponto de trabalho mais 1,0 ponto de AAI*).
- A atividade deve se iniciar no laboratório e ser entregue pelo SINEF **até dia 13/nov/2012**.
- Deve ser entregue um arquivo compactado chamado **`Ativ09.zip`** com um arquivo chamado **`grupo.txt`** contendo o nome dos integrantes e os arquivos **`c`** cujos nomes devem seguir o seguinte padrão: **`Ativ09Ex01.c, Ativ09Ex02.c,...`** **Isto será avaliado!**
- Apenas **um** do grupo deverá entregar. **Isto também será avaliado!**
- Atividade não serão recebidas/pontuadas **após o prazo de entrega**. Portanto, fique atento ao prazo.
- **Cópias** (total ou parcial) serão penalizadas com **nota zero**.
- A atividade é em grupo de, no máximo, **dois** alunos.

Atividade 09 - Lista de Exercícios – Argumentos, União, Enumeração e Diretivas

1. Criar um programa C ANSI chamado **`matematica`** que tenha que receber três parâmetros. O primeiro trata-se da operação operação a ser realizada (`somar, subtrair, multiplicar, dividir`) e os outros dois parâmetros tratam-se de dois números de ponto flutuante. Depois irá somente exibir o resultado da operação. Caso não sejam passados os parâmetros ou eles sejam incorretos, deverá ser exibido ao usuário como utilizar o programa.
   Ex:
   `> matematica.exe somar 2.6 4.4` → `7.0`
   `> matematica.exe sub 300.0 4.2` → *`Informará como utilizar o programa`*

   Dica: Utilize a função **`atof`** para converter *string* para *double*

2. Desenvolva a função **`ler`** e **`exibir`** do programa C ANSI abaixo.
   A função **ler** deverá ler o nome do aluno e a nota como inteiro e se for:
   - entre 90 e 100, atribuirá a enumeração **`A`**
   - entre 80 e 89, **`B`**
   - entre 70 e 79, **`C`**
   - entre 60 e 69, **`D`**
   - entre 0 a 59, **`E`**

   Observe que você atribui **`A`** à nota do aluno e não o caractere **`'A'`**.

   A função **exibir** deverá exibir as informações do aluno de acordo com a nota:
   - A → *"Excelente #nomeAluno#!"*
   - B → *"Ótimo #nomeAluno#!"*
   - C e D → *"Caro #nomeAluno#, fez nada além de sua obrigação."*
   - E → "Caro *#nomeAluno#, você foi reprovado."*

   Dica: Observe que não é o caractere 'A', mas sim

```
typedef enum {A = 1, B, C, D, E} nota;

typedef struct {
    char nomeAluno[60];
    nota n;
}aluno;

void ler(aluno *a);
void exibir(aluno *a);

int main(int argc, char *argv[]){
    aluno a;
    ler(&a);
    exibe(&a);
    return 0;
}
```

3. Observe a representação binária do número inteiro 33.489.761:

   ```
   0000 0001   1111 1111   0000 0011   0110 0001
   ```

   Crie um programa C ANSI que atribua o número acima ao inteiro **`i`** da união abaixo, depois imprima o inteiro curto **`s`** e, por fim, imprima o caractere **`ch`**.

   ```
   typedef union {
       int i;
       short int s;
       char ch;
   } u_num;
   ```

   Explique o resultado e desenhe como as variáveis compartilham o mesmo espaço de memória. Responda como comentário no próprio código fonte.

4. Crie um programa C ANSI que o programador possa **também** escrever:
   - **`imprime`** para **`printf`**
   - **`escreve`** para **`scanf`**
   - **`enquanto`** para **`while`**
   - **`para`** para **`for`**
   - **`se`** para **`if`**
   - **`senao`** para **`else`**

   Além disso:
   - A constante **`PI`** deverá ser definida com **`3.141596`**;
   - Se o programa estiver em testes haja um macro teste definida e todo teste deve ser feito dentro do bloco:

     ```
     #ifdef teste
        ...
     #endif
     ```

   O programa deve inicializar todo o arranjo com valores aleatórios de zero até o tamanho. E caso o programa esteja em testes:
   - Deve ser exibido qual foi o valor aleatório para cada posição do arranjo antes da atribuição;
   - Após as atribuições, deverá ser exibido qual foi o valor que mais saiu e quantas vezes.
     - Por exemplo, o valor 5 foi escolhido 12 vezes.

5. [LÓGICA] Faça um programa de *login* que receba uma senha alfanumérica de exatamente 8 caracteres e retorne um número inteiro longo correspondente. Dica: Use união.

6. [LÓGICA] Altere a **`Ativ06Ex09`** de forma que receba o arranjo por argumento de linha de comando e que funcione para qualquer tamanho de arranjo.