

PROJETO 1 - CALCULADORA

1. Descrição Geral

- Elaborar um programa em linguagem C que execute as operações básicas de SOMA e SUBTRAÇÃO para uma máquina com precisão pré estabelecida.
- Elaborar um relatório cujo conteúdo deve abranger: o conhecimento teórico codificado no programa, as especificações de como o código foi realizado, exemplos de operações realizadas pelo programa, discussão sobre os resultados, e limitações/dificuldades do projeto.
- Preparar-se para uma avaliação oral sobre o projeto. Esta avaliação será definida por cada professor.

2. Equipes

- Os alunos da mesma turma, poderão formar equipes de 1 até 5 componentes. A equipe deverá enviar e-mail para o professor até a data 16/09/2016, de maneira a confirmar a inscrição da equipe. Um representante da equipe deve enviar e-mail, com o nome completo e CPF de todos os integrantes da equipe.

3. Datas Importantes

- 23/08/2016 - Lançamento do projeto.
- 16/09/2016 - Limite para inscrição das equipes.
- 27/09/2016 - Entrega do projeto.

4. O que deverá ser entregue

- Envelope com o nome de todos os integrantes da equipe, o relatório e código fonte do programa.
- Arquivo executável do programa (trazer em um pen-drive para que o professor possa gravá-lo). A equipe deve se assegurar que o executável do programa esteja funcionando (rodando em qualquer computador).
- A entrega deve ser feita no início da aula desta data. O professor só receberá o projeto neste período, e apenas se os três itens forem entregues (relatório + código fonte + arquivo executável).

5. O Relatório

- O relatório deve ser escrito de acordo com o modelo disponível no site da disciplina: <http://www.cin.ufpe.br/~if215>.
- A equipe deve elaborá-lo de maneira a abranger: o conhecimento teórico codificado no programa, as especificações de como o código foi realizado, exemplos de operações realizadas pelo programa, discussão sobre os resultados, e limitações/dificuldades do projeto.
- Exemplos de cálculos realizados com o programa são **essenciais**. De preferência, deve-se variar a precisão da Máquina. Em todos os casos, os exemplos devem ser **analisados**.

6. Avaliação do Projeto

- A nota do projeto é uma composição ponderada de três notas: 25% do programa; 25% do relatório; 50% da avaliação presencial. Esta nota terá peso de 40% sobre a 1ª nota (60% prova 1 + 40% projeto 1).

- A avaliação presencial será realizada por um representante do grupo. A escolha deste representante assim como a data da avaliação presencial, serão definidas por cada professor.

7. Observações sobre o projeto

- Não será admitida cópia, parcial ou total, de códigos e/ou relatórios. Caso isso ocorra com alguma equipe, a nota do projeto será zerada.
- As Especificações do Programa devem ser obedecidas para a implementação do programa. O programa codificado sem as especificações descritas a seguir será zerado.

8. Especificações do Programa:

- **Ideia Geral:** O programa receberá um arquivo com a especificação da máquina $F(10, t, e_{min}, e_{max})$ (como visto em aula) e uma série de operações a serem realizadas. Estas operações deverão seguir os algoritmos vistos em sala. É proibido a utilização das operações aritméticas pré-definidas na linguagem C, isto é, as operações devem ser implementadas como em uma calculadora: trabalhando com cada dígito individualmente.
- **Especificações do arquivo de entrada:** O arquivo de entrada será da seguinte forma:

$$\begin{aligned}
 &t; e_{min}; e_{max}; N \\
 &x_1^1; x_2^1; \text{operação}_1 \\
 &x_1^2; x_2^2; \text{operação}_2 \\
 &\dots \\
 &x_1^N; x_2^N; \text{operação}_N
 \end{aligned}$$

Onde, N especifica o número de operações a ser realizadas e x_j^i é um número com mantissa normalizada com 7 dígitos significativos. Além disso, utilizaremos os seguintes limites para a especificação da máquina $2 \leq t \leq 7$, $-9 \leq e_{min} \leq -4$ e $4 \leq e_{max} \leq 9$. Um exemplo de um arquivo de entrada, chamado *input.dat* pode ser visto abaixo:

```

4;-6;+6;5
+4.345670E+01;+2.125000E+00;+
+1.000000E-02;-4.342107E+03;-
+6.590472E-01;+2.771043E+07;+
-5.003500E-02;-8.679890E-02;-
+2.083090E-05;+2.081400E-05;-

```

O significado das linhas está descrito a seguir:

- Linha 1: n° de significativos, menor expoente, maior expoente, quantidade de operações matemáticas que devem ser realizadas neste arquivo (separados por ”;”);
- Linha 2 em diante: primeira parcela (sinal, mantissa normalizada com seis casas decimais, seguido do expoente), segunda parcela, operação de SOMA ou de SUBTRAÇÃO (as três informações estão separadas por ”;”)
- **Regiões de operação da calculadora:**
 - A máquina deve acusar se qualquer parcela ou se o resultado de uma operação estiver fora da região de operação da máquina, identificando-a entre *undeflow* e *overflow*.
- O menu de apresentação do programa deve apresentar as opções: i) Digitar o nome do arquivo de entrada; ii) ou Sair do programa.
- A saída deverá ser impressa na tela e deve seguir a representação da Máquina F. Serão impressas na tela: i) as informações da máquina; ii) as informações de entrada com os respectivos resultados das operações. Para o exemplo do arquivo ”*input.dat*” a impressão na tela será da seguinte forma:

```

F(10,4,-6,+6)
n° de oper.: 5
+4.346E+01;+2.125E+00;+; resultado: +4.558E+01
+1.000E-02;-4.342E+03;-; resultado: -4.342E+03
+6.590E-01;+2.771E+07;+; resultado: parcela 2 inválida, overflow
-5.004E-02;-8.680E-02;-; resultado: +3.676E+03
+2.083E-05;+2.081E-05;-; resultado: underflow

```

- Definição das variáveis locais:
 - É proibido declarar variáveis reais (FLOAT ou DOUBLE). O programa deve ter exclusivamente variáveis do tipo INT ou CHAR (SHORT e LONG INT também estão proibidas).
- Representação dos números no programa:
 - Um número necessita de três variáveis para representá-lo: sinal, expoente e mantissa. A mantissa será representada por um vetor de inteiros de dimensão 15 e deverá estar normalizada (o primeiro elemento do vetor deve ser sempre um dígito diferente de "0").
 - De acordo com o item anterior, serão aceitas duas formas de representar um número: i) definindo três variáveis para armazenar o sinal, o expoente e a mantissa normalizada (vetor de dimensão 15); ii) ou definindo um único vetor de dimensão 17 que inclui as três variáveis.
 - Como exemplo, o número +6.875408E-03 poder ser representado pelos seguintes modelos (seguindo a sintaxe padrão da linguagem C):

```

Modelo 1
n° de oper.: 5
INT sinal = +1
INT expoente = -3
INT mantissa[15] = { 6, 8, 7, 5, 4, 0, 8 }

```

```

Modelo 2
INT numero[17] = { 1, -3, 6, 8, 7, 5, 4, 0, 8 }

```

- Função de arredondamento:
 - Deve ser implementada exatamente como ensinada em sala de aula (Dias & Zanoni).

BIBLIOGRAFIA

- Santos & Silva, Métodos Numéricos 3ª Edição, Editora Universitária UFPE, 2010.