Aula Prática 2

Objetivos:

- Máximo Divisor Comum

Versão Inicial: 30/04/2024

Prazo: 06/05/2024 – 08:00

Observações:

- Leia este enunciado com **MUITA** atenção até o final antes de iniciar o trabalho.
- Os arquivos solicitados deverão estar disponíveis nos diretórios correspondentes (***Aulas-Praticas*** e ***RCS***) até o prazo estipulado acima. Cuidado com os nomes dos diretórios e dos arquivos. Deverão ser exatamente os definidos neste roteiro (incluindo maiúsculas, minúsculas, caracteres especiais e extensões, se existentes).
- **As tarefas deverão ser executadas na ordem solicitada neste roteiro.**
- O padrão de nomenclatura definido em aula para os identificadores deverá ser utilizado:
  - ***Snake Case***
    - letras maiúsculas, dígitos e o caractere sublinhado;
    - macros e constantes;
    - Exemplo: ***COMPRIMENTO_MAXIMO_NOME***
  - ***Camel Case***
    - letras minúsculas, com exceção do primeiro caractere de cada palavra e a partir da segunda palavra que compõe o identificador;
    - tipos, membros de tipos enumerados e variáveis;
    - Exemplo: ***comprimentoNomeCompleto***
  - ***Pascal Case***
    - letras minúsculas, com exceção do primeiro caractere de cada palavra que compõe o identificador;
    - funções
    - Exemplo: ***CalcularFatorial***
  - Todos os identificadores podem conter dígitos, mas nenhum identificador pode começar com dígito.
  - Todos os identificadores devem ser significativos e sem abreviações.
  - Identificadores de função devem começar com um verbo no infinitivo.
- Os arquivos de dependências deverão possibilitar que a compilação e que a *linkedição* sejam executadas utilizando-se tanto o ***gcc***, quanto o ***clang***. A seleção da ferramenta utilizada deverá ser realizada no momento da execução do comando ***make***. O ***gcc*** deverá ser considerado como o valor padrão para a ferramenta de compilação e de *linkedição*.

  Para a definição da ferramenta desejada, deverá ser utilizada uma macro no *FreeBSD* e um argumento com o valor desejado no *Linux*.

  As duas macros utilizadas deverão ser ***GCC*** e ***CLANG*** (definidas usando a opção de linha de comando **-D** do comando ***make***).

  O argumento, identificado por ***cc***, deverá ser igual a ***GCC*** ou a ***CLANG***.
- Independente da ferramenta utilizada para a compilação, as opções de compilação poderão ser redefinidas no instante da execução do comando ***make*** (mantendo-se a exibição de todas as mensagens de advertência, definida pelo valor ***-Wall***). O valor padrão para estas opções deverá ser ***-Wall -ansi***.

Estas opções poderão ser redefinidas através de macros ou através de argumentos (de forma semelhante àquela utilizada para definir o compilador/*linkeditor*).

No *FreeBSD* deverão ser definidas as macros ***ANSI***, ***C89***, ***C90***, ***C99*** e ***C11.***

No *Linux* deverá ser definido o argumento ***dialeto*** com um dos seguintes valores: ***ANSI***, ***C89***, ***C90***, ***C99*** ou ***C11***.

- Os arquivos de dependências deverão incluir a macro ***DIALECT*** contendo o dialeto a ser utilizado na compilação do código. Esta macro será inicialmente igual a ***ansi*** e poderá ser alterada para ***c89***, ***c90***, ***c99*** ou ***c11*** de acordo com o esquema definido acima.
- Os arquivos de dependências deverão incluir também a macro ***STANDARD*** contendo a opção de linha de comando correspondente ao dialeto selecionado. Se, por exemplo, o dialeto selecionado for o ***ANSI***, esta macro deverá ser igual a ***-ansi***. Por outro lado, se o dialeto for uma das outras quatro opções, esta macro deverá ser igual a ***-std=CXX***, onde ***XX*** deverá ser substituído pelo número correspondente (se o dialeto for igual a ***C89***, *XX* deverá ser igual a ***89***, se o dialeto for igual a ***C90***, ***XX*** deverá igual a ***90*** e assim por diante).
- A *linkedição* deverá utilizar a opção ***-Wall***.
- Cuidado com os nomes das macros e dos rótulos. Deverão ser exatamente os definidos neste roteiro (maiúsculas, minúsculas, caracteres especiais e extensões, se existentes).
- Todos os rótulos solicitados no roteiro são obrigatórios. Durante a correção, caso não seja possível alcançar os objetivos (binários e/ou bibliotecas e limpezas de código) solicitados, a nota correspondente ao item/aula em questão será igual a zero.
- Seguem alguns exemplos (todos devem funcionar):
  - ***make*** - compila/*linkedita* (tanto no *FreeBSD*, quanto no *Linux*) com a ferramenta e dialeto padrões, ou seja, ***gcc*** e **ANSI** respectivamente.
  - ***make clean-all all***
  - ***make clean-all aula01***
  - ***make clean aula0101***
  - ***make -DGCC*** - compila/*linkedita* usando o ***gcc*** e o dialeto **ANSI** (somente *FreeBSD*).
  - ***make -DCLANG*** - compila/*linkedita* usando o ***clang*** e o dialeto **ANSI** (somente *FreeBSD*).
  - ***make cc=GCC*** - compila/*linkedita* usando o ***gcc*** e o dialeto **ANSI** (somente *Linux*).
  - ***make cc=CLANG*** - compila/*linkedita* usando o ***clang*** e o dialeto **ANSI** (somente *Linux*).
  - ***make -DCLANG -DC89*** - compila/*linkedita* usando o ***clang*** e o dialeto **C89** (somente *FreeBSD*).
  - ***make -DCLANG -DC11*** - compila/*linkedita* usando o ***clang*** e o dialeto **C11** (somente *FreeBSD*).
  - ***make cc=CLANG dialeto=C99*** - compila/*linkedita* usando o ***clang*** e o dialeto **C99** (somente *Linux*).
  - ***make cc=GCC dialeto=C90*** - compila/*linkedita* usando o ***gcc*** e o dialeto **ANSI** (somente *Linux*).

- Inclua, no início de todos os arquivos solicitados (código-fonte e arquivos de dependências), os seguintes comentários (**sem caracteres especiais**):

```
Universidade Federal do Rio de Janeiro
Escola Politecnica
Departamento de Eletronica e de Computacao
EEL270 - Computacao II - Turma 2024/1
Prof. Marcelo Luiz Drumond Lanza
Autor: <nome completo>
Descricao: <descrio sucinta dos objetivos do programa>
$Author$
$Date$
$Log$
```

- Inclua, no final de todos os arquivos solicitados, o seguinte comentário:

```
$RCSfile$
```

O máximo divisor comum (MDC) entre dois números inteiros não negativos X e Y pode ser definido como:

MDC (X,Y) = MDC (Y,Z) se Z (resto da divisão de X por Y) é diferente de zero.
MDC (X,Y) = Y se Z é igual a zero.
MDC (X,Y) = X se Y é igual a zero.
MDC (X,Y) = Y se X é igual a zero.
MDC (X,Y) = 0 se X e Y é igual a zero (indicando condição de erro).

1. Crie, baseado na definição acima, o arquivo ***aula0201.h*** contendo a definição do tipo ***ull*** (correspondendo a ***unsigned long long***) e a definição do protótipo da função ***CalcularMaximoDivisorComum***. Esta função deverá receber dois números inteiros não negativos e deverá retornar o valor do máximo divisor comum destes números. O identificador da macro referente à combinação ***ifndef*** e ***define*** deverá ser igual a ***AULA0201.*** Esta macro deverá ser definida e deverá ser definida como uma *string* igual a "@(#)aula0201.h $Revision$".

   ```
   ull
   CalcularMaximoDivisorComum (ull, ull);
   ```

2. Crie o arquivo ***aula0201a.c*** contendo a implementação da função solicitada no item anterior. Esta implementação deverá utilizar recursividade.

3. Crie o arquivo ***aula0202.c*** contendo a implementação de um programa de testes para a função ***CalcularMaximoDivisorComum***. Este programa deverá receber dois números inteiros não negativos através dos argumentos de linha de comando (CLI) e deverá exibir o valor do máximo divisor comum dentre estes dois números. Todos os tratamentos de erro necessários e que não puderem ser realizados na função ***CalcularMaximoDivisorComum*** deverão ser implementados neste programa. A saída gerada por este programa deverá exibir o MDC entre os dois números recebidos (no formato abaixo e na cor verde). Caso contrário deverá exibir a mensagem de erro correspondente (na cor vermelha).

   MDC (24, 16) = 8

4. Inclua, nos arquivos de dependências, as macros ***AULA0202AOBJS*** (correspondendo aos arquivos ***aula0201a.o*** e ***aula0202.o***) e ***AULA02*** (correspondendo inicialmente ao executável ***aula0202a***). Altere o valor da macro ***EXECS***, de forma que inclua o valor da macro ***AULA02***. Inclua também os objetivos ***aula02*** e ***aula0202a*** com os comandos correspondentes.

5. Gere e teste as 20 versões do executável ***aula0202a***.

6. Submeta os arquivos ***aula0201.h***, ***aula0201a.c***, ***aula0202.c, BSDmakefile*** e ***GNUmakefile*** ao sistema de controle de versão.

7. Recupere uma cópia de leitura dos arquivos ***aula0201a.c*** e ***aula0202.c*** e uma cópia de escrita do arquivo ***aula0201.h*** e dos arquivos de dependências.

8. Crie as macros ***LIBMATEMATICARECURSAOOBJS*** e ***LIBMATEMATICARECURSAO,*** correspondendo respectivamente ao arquivo ***aula0201a.o*** e à biblioteca ***libmatematica-recursao.a***.
9. Gere o arquivo ***libmatematicarecursao.a***.
10. Crie o arquivo ***aula0201b.c*** contendo a implementação da função definida no item 1. Esta implementação deverá utilizar o laço de repetição do ... while.
11. Inclua, nos arquivos de dependências, a macro ***AULA0202BOBJS*** (correspondendo aos arquivos ***aula0201b.o*** e ***aula0202.o***) e o objetivo ***aula0202b*** com os comandos correspondentes. Altere o valor da macro ***AULA02***, incluindo o executável correspondente.
12. Gere e teste as 20 versões do executável ***aula0202b***.
13. Submeta os arquivos ***aula0201.h***, ***aula0201b.c***, ***BSDmakefile*** e ***GNUmakefile*** ao sistema de controle de versão.
14. Recupere uma cópia de leitura dos arquivos contendo o código-fonte e uma cópia de escrita dos arquivos de dependências.
15. Crie as macros ***LIBMATEMATICADOWHILEOBJS*** e ***LIBMATEMATICADOWHILE,*** correspondendo respectivamente ao arquivo ***aula0201b.o*** e à biblioteca ***libmatematica-dowhile.a***.
16. Gere o arquivo ***libmatematicadowhile.a***.
17. Crie o arquivo ***aula0201c.c*** contendo a implementação da função definida no item 1. Esta implementação deverá utilizar o laço de repetição for.
18. Inclua, nos arquivos de dependências, a macro ***AULA0202COBJS*** e o objetivo ***aula0202c*** com os comandos correspondentes. Altere o valor da macro ***AULA02*** incluindo o executável correspondente.
19. Gere e teste as 20 versões do executável **aula0202c**.
20. Submeta os arquivos ***aula0201c.c, BSDmakefile*** *e* ***GNUmakefile*** ao sistema de controle de versão.
21. Recupere uma cópia de leitura dos arquivos contendo o código-fonte e uma cópia de escrita dos arquivos de dependências.
22. Crie as macros ***LIBMATEMATICAFOROBJS*** e ***LIBMATEMATICAFOR,*** correspondendo respectivamente ao arquivo ***aula0201c.o*** e à biblioteca ***libmatematica-for.a***.
23. Gere o arquivo ***libmatematicafor.a***.
24. Crie o arquivo ***aula0201d.c*** contendo a implementação da função definida no item 1. Esta implementação deverá utilizar o laço de repetição while.
25. Inclua, nos arquivos de dependências, a macro ***AULA0202DOBJS*** e o objetivo ***aula0202d*** com os comandos correspondentes. Altere o valor da macro ***AULA02*** incluindo o binário correspondente.
26. Gere e teste as 20 versões do executável ***aula0202d***.

27. Submeta o arquivo ***aula0201d.c, BSDmakefile*** *e* ***GNUmakefile*** ao sistema de controle de versão.
28. Recupere uma cópia de leitura dos arquivos contendo o código-fonte e uma cópia de escrita dos arquivos de dependências.
29. Crie as macros ***LIBMATEMATICAWHILEOBJS*** e ***LIBMATEMATICAWHILE,*** correspondendo respectivamente ao arquivo ***aula0201d.o*** e à biblioteca ***libmatematica-while.a***.
30. Gere o arquivo ***libmatematicawhile.a***.
31. Submeta os arquivos ***BSDmakefile*** *e* ***GNUmakefile*** ao sistema de controle de versão.
32. Recupere uma cópia de escrita dos arquivos de dependências.
33. Limpe o diretório (*make clean-all*).
34. Arquivos que devem ser disponíveis ao final da aula:

    Subdiretório "~/private/EEL270/2023-2/Aulas-Praticas"
    - *aula0201.h*
    - *aula0201a.c*
    - *aula0201b.c*
    - *aula0201c.c*
    - *aula0201d.c*
    - *aula0202.c*
    - *BSDmakefile*
    - *GNUmakefile*

Além dos correspondentes gerados pela ferramenta de controle de versão (localizados no subdiretório ***RCS***) e dos arquivos gerados nas aulas práticas anteriores.