

Aula Prática 4 - Roteiro

20/04/2021 - Roteiro referente à aula prática 4 - Potênciação/Serie Harmônica Alternada

Versões:

- 17/08/2021 - Versão inicial

Prazo: 21/08/2021 - 18:00

Valor: 10,0 - Peso: 2

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-Práticas e RCS**) até o prazo estipulado acima. Cuidado com os nomes dos diretórios e dos arquivos. Deverão ser exatamente os definidos neste roteiro (maiúsculas, minúsculas, caracteres especiais e extensões, se existentes).
- **As tarefas deverão ser executadas na ordem solicitada neste roteiro.**
- **Os itens marcados com (*) deverão ser entregues até às 19:00 do dia da aula em questão.**
- Os arquivos de dependências deverão possibilitar que a compilação e 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 o valor padrão para a compilação e para a *linkedição*.

Para a definição da ferramenta desejada deverá ser utilizada uma macro (no *FreeBSD*) ou um argumento com o valor desejado (no *CentOS*). As duas macros utilizadas deverão ser *GCC* e *CLANG* (definidas usando a opção *-D*). O argumento, identificado por *cc*, deverá ser igual a *GCC* ou *CLANG*.

- Independente da ferramenta utilizada para a compilação, o *flag* de compilação deverá ser definido no instante da execução do comando *make*. O valor padrão para este *flag* deverá ser *"-Wall -ansi"* (sem as aspas).

Durante a execução do comando *make* poderão ser definidos outros valores para este *flag* (mantendo a opção de exibir todas as mensagens de advertência) 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*, enquanto que no *CentOS* deverá ser definido o argumento *dialeto* com os valores *ANSI*, *C89*, *C90*, *C99* ou *C11*.

- Crie uma macro, *DIALETO*, contendo o dialeto a ser utilizado na compilação do código. Esta macro será inicialmente igual a *"ansi"* e poderá ser alterada para *"c89"*, *"c99"* ou *"c11"* de acordo com o esquema definido acima.
- O *flag* de *linkedição* deverá ser igual a *"-Wall"* (sem as aspas).
- 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 questão será zero.
- Seguem alguns exemplos:

make - compila/linkedita (tanto no *FreeBSD*, quanto no *CentOS*) com a ferramenta e dialeto padrões, ou seja, *gcc* e *ANSI* respectivamente.

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 CentOS).
 make cc=CLANG - compila/linkedita usando o *clang* e o dialeto ANSI (somente CentOS).
 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 CentOS).

- Inclua, no início de todos os arquivos solicitados, os seguintes comentários:

```
Universidade Federal do Rio de Janeiro
Escola Politecnica
Departamento de Eletronica e de Computacao
EEL270 - Computacao II - Turma 2021/1
Prof. Marcelo Luiz Drumond Lanza
Autor: <nome completo>
Descricao: <descrição sucinta dos objetivos do programa>
```

```
$Author$
$Date$
$Log$
```

- Inclua, no final de todos os arquivos solicitados, os seguintes comentários:

```
$RCSfile$
```

1. (*) Crie o arquivo "*aulao4.h*" contendo o protótipo da função *CalcularExponencial*. Esta função deverá receber uma base (do tipo real) e um expoente (do tipo inteiro) e deverá retornar o valor correspondente à base elevada ao expoente. Lembre-se que o expoente pode ser tanto negativo, quanto positivo.

Considere que 0^0 é igual a 1 e que o elevado a números negativos é igual a infinito.

A macro referente à combinação *ifndef* e *define*, por exemplo *_AULA04_*, deverá ser definida como uma *string* igual a:

```
"@(#)aulao4.h $Revision$"
```

O protótipo da função é definido a seguir:

```
long double
CalcularExponencial (double, int);
```

2. (*) Crie o arquivo "*aulao401a.c*" contendo a implementação da função *CalcularExponencial*. Esta implementação deverá utilizar recursividade e não poderá utilizar nenhuma função de nenhuma biblioteca padrão.
3. (*) Crie o arquivo "*aulao402.c*" contendo a implementação de um programa de testes para a função *CalcularExponencial*. Este programa deverá receber dois valores através de argumentos da CLI, correspondendo respectivamente à base e ao expoente desejados. Para a conversão de *string* em real utilize a função *strtod* e de *string* em inteiro utilize a função *strtol*.
4. (*) Inclua, nos arquivos de dependências, as macros *AULA0402AOBJS* e *AULA04*. Altere o valor da macro *EXECS*, de forma que inclua o valor da macro *AULA04*. Inclua também os objetivos *aulao4* e *aulao402a* com os comandos correspondentes.
5. (*) Gere e teste as 16 versões do executável *aulao402a*.
6. (*) Submeta os arquivos "*aulao4.h*", "*aulao401a.c*", "*aulao402.c*" e "**makefile*" ao sistema de controle de versão.
7. (*) Recupere uma cópia de leitura dos arquivos "*aulao401a.c*" e "*aulao402.c*" e uma cópia de escrita dos arquivos "*aulao4.h*" e "**makefile*".
8. (*) Adicione o arquivo "*aulao401a.o*" ao valor da macro *LIBMATEMATICARECURSAOOBJS*.
9. (*) Gere o arquivo "*libmatematicarecursao.a*".
10. Crie o arquivo "*aulao401b.c*" contendo a implementação da função *CalcularExponencial*. Esta implementação deverá utilizar o laço de repetição *do ... while* e não poderá utilizar nenhuma função de nenhuma biblioteca padrão.
11. Inclua, nos arquivos de dependências, a macro *AULA0402BOBJS* e o objetivo *aulao402b* com os

comandos correspondentes. Altere o valor da macro *AULA04*, incluindo o binário correspondente.

12. Gere e teste as 16 versões do executável *aulao402b*.
13. Submeta os arquivos "*aulao401b.c*" e "**makefile*" ao sistema de controle de versão.
14. Recupere uma cópia de leitura do arquivo "*aulao401b.c*" e uma cópia de escrita dos arquivos "**makefile*".
15. Adicione o arquivo "*aulao401b.o*" ao valor da macro *LIBMATEMATICADOWHILEOBS*.
16. Gere o arquivo "*libmatematicadowhile.a*".
17. Crie o arquivo "*aulao401c.c*" contendo a implementação da função *CalcularExponencial*. Esta implementação deverá utilizar o laço de repetição *for* e não poderá utilizar nenhuma função de nenhuma biblioteca padrão.
18. Inclua, nos arquivos de dependências, a macro *AULA0402COBS* e o objetivo *aulao402c* com os comandos correspondentes. Altere o valor da macro *AULA04*, incluindo o binário correspondente.
19. Gere e teste as 16 versões do executável *aulao402c*.
20. Submeta os arquivos "*aulao401c.c*" e "**makefile*" ao sistema de controle de versão.
21. Recupere uma cópia de leitura do arquivo "*aulao401c.c*" e uma cópia de escrita dos arquivos "**makefile*".
22. Adicione o arquivo "*aulao401c.o*" ao valor da macro *LIBMATEMATICAFOROBS*.
23. Gere o arquivo "*libmatematicafor.a*".
24. Crie o arquivo "*aulao401d.c*" contendo a implementação da função *CalcularExponencial*. Esta implementação deverá utilizar o laço de repetição *while* e não poderá utilizar nenhuma função de nenhuma biblioteca padrão.
25. Inclua, nos arquivos de dependências, a macro *AULA0402DOBS* e o objetivo *aulao402d* com os comandos correspondentes. Altere o valor da macro *AULA04*, incluindo o binário correspondente.
26. Gere e teste as 16 versões do executável *aulao402d*.
27. Submeta os arquivos "*aulao401d.c*" e "**makefile*" ao sistema de controle de versão.
28. Recupere uma cópia de leitura do arquivo "*aulao401d.c*" e uma cópia de escrita dos arquivos "**makefile*".
29. Adicione o arquivo "*aulao401d.o*" ao valor da macro *LIBMATEMATICAWHILEOBS*.
30. Gere o arquivo "*libmatematicawhile..a*".
31. Inclua, no arquivo "*aulao4.h*", o protótipo da função *CalcularSerieHarmonicaAlternada*. Esta função deverá receber um número inteiro não negativo representando o número de termos que deverá ser utilizado para calcular o valor da série harmônica alternada. A função deverá retornar o valor calculado.

Considere que:

$S(n) = 0$ se n é igual a 0.

$S(n) = 1 - 1/2^2 + 1/3^3 - 1/4^4 + \dots - 1/n^n$ se n é par

$S(n) = 1 - 1/2^2 + 1/3^3 - 1/4^4 + \dots + 1/n^n$ se n é ímpar

O protótipo da função é definido a seguir:

float

CalcularSerieHarmonicaAlternada (unsigned short int);

32. Crie o arquivo "*aulao403a.c*" contendo a implementação da função *CalcularSerieHarmonicaAlternada*. Esta implementação deverá utilizar recursividade e a função *CalcularExponencial* (através da biblioteca "*libmatematicarecursao.a*"). Esta implementação não poderá utilizar nenhuma função de nenhuma biblioteca padrão.
33. Crie o arquivo "*aulao404.c*" contendo a implementação de um programa de testes para a função *CalcularSerieHarmonicaAlternada*. Este programa deverá receber um valor maior do que 0 e menor do que 1 através de argumentos da CLI. O programa deverá exibir o valor da série variando n de 0 até N (sempre com 10 casas decimais), de forma que $|S(N) - S(N-1)|$ seja menor ou igual a P vezes $S(N-1)$, onde P é o percentual recebido através dos argumentos da CLI. Para a conversão do argumento de linha de comando em real utilize a função *strtod*.
34. Inclua, nos arquivos de dependências, a macro *AULA0404AOBS* e o objetivos *aulao404a* com os comandos correspondentes. Altere o valor da macro *AULA04*, incluindo o binário correspondente.

Para usar uma biblioteca estática inclua, no final da linha de linkedição, o seguinte: -L.
-INOMEDABIBLIOTECA, onde NOMEDABIBLIOTECA corresponde ao nome do arquivo correspondente sem o prefixo "lib" e o sufixo ".a".

35. Gere e teste as 16 versões do executável *aulao404a* utilizando a biblioteca criada (correspondente).
36. Submeta os arquivos "*aulao4.h*", "*aulao403a.c*", "*aulao404.c*" e "**makefile*" ao sistema de controle de versão.
37. Recupere uma cópia de leitura dos arquivos "*aulao4.h*", "*aulao403a.c*" e "*aulao404.c*" e uma cópia de escrita dos arquivos e "**makefile*".
38. Adicione o arquivo "*aulao403a.o*" ao valor da macro *LIBMATEMATICARECURSAO**OBJS*.
39. Gere o arquivo "*libmatematicarecursao.a*".
40. Crie o arquivo "*aulao403b.c*" contendo a implementação da função *CalcularSerieHarmonicaAlternada*. Esta implementação deverá utilizar o laço de repetição *do ... while* e a função *CalcularExponencial* (através da biblioteca "*libmatematicadowhile.a*"). Esta implementação não poderá utilizar nenhuma função de nenhuma biblioteca padrão.
41. Inclua, nos arquivos de dependências, a macro *AULA0404BOBJS* e o objetivo *aulao404b* com os comandos correspondentes. Altere o valor da macro *AULA04*, incluindo o binário correspondente.
42. Gere e teste as 16 versões do executável *aulao404b* utilizando a biblioteca criada (correspondente).
43. Submeta os arquivos "*aulao403b.c*" e "**makefile*" ao sistema de controle de versão.
44. Recupere uma cópia de leitura do arquivo "*aulao403b.c*" e uma cópia de escrita dos arquivos "**makefile*".
45. Adicione o arquivo "*aulao403b.o*" ao valor da macro *LIBMATEMATICADOWHILE**OBJS*.
46. Gere o arquivo "*libmatematicadowhile.a*".
47. Crie o arquivo "*aulao403c.c*" contendo a implementação da função *CalcularSerieHarmonicaAlternada*. Esta implementação deverá utilizar o laço de repetição *for* e a função *CalcularExponencial* (através da biblioteca "*libmatematicafor.a*"). Esta implementação não poderá utilizar nenhuma função de nenhuma biblioteca padrão.
48. Inclua, nos arquivos de dependências, a macro *AULA0404COBJS* e o objetivo *aulao404c* com os comandos correspondentes. Altere o valor da macro *AULA04*, incluindo o binário correspondente.
49. Gere e teste as 16 versões do executável *aulao404c* utilizando a biblioteca criada (correspondente).
50. Submeta os arquivos "*aulao403c.c*" e "**makefile*" ao sistema de controle de versão.
51. Recupere uma cópia de leitura do arquivo "*aulao403c.c*" e uma cópia de escrita dos arquivos "**makefile*".
52. Adicione o arquivo "*aulao403c.o*" ao valor da macro *LIBMATEMATICAFOR**OBJS*.
53. Gere o arquivo "*libmatematicafor.a*".
54. Crie o arquivo "*aulao403d.c*" contendo a implementação da função *CalcularSerieHarmonicaAlternada*. Esta implementação deverá utilizar o laço de repetição *while* e a função *CalcularExponencial* (através da biblioteca "*libmatematicawhile.a*"). Esta implementação não poderá utilizar nenhuma função de nenhuma biblioteca padrão.
55. Inclua, nos arquivos de dependências, a macro *AULA0404DOBJS* e o objetivo *aulao404d* com os comandos correspondentes. Altere o valor da macro *AULA04*, incluindo o binário correspondente.
56. Gere e teste as 16 versões do executável *aulao404d* utilizando a biblioteca criada (correspondente).
57. Submeta os arquivos "*aulao403d.c*" e "**makefile*" ao sistema de controle de versão.
58. Recupere uma cópia de leitura do arquivo "*aulao403d.c*" e uma cópia de escrita dos arquivos "**makefile*".
59. Adicione o arquivo "*aulao403d.o*" ao valor da macro *LIBMATEMATICAWHILE**OBJS*.
60. Gere o arquivo "*libmatematicawhile.a*".
61. Crie o arquivo "*aulao403e.c*" contendo a implementação da função *CalcularSerieHarmonicaAlternada*. Esta implementação deverá utilizar recursividade ou o laço de repetição desejado e a função da biblioteca *Math* mais adequada para o cálculo da exponencial necessária.
62. Inclua, nos arquivos de dependências, a macro *AULA0404EOBJS* e o objetivo *aulao404e* com os comandos correspondentes. Altere o valor da macro *AULA04*, incluindo o binário correspondente.
63. Gere e teste as 16 versões do executável *aulao404e*.

64. Submeta os arquivos "*aulao403e.c*" e "**makefile*" ao sistema de controle de versão.
65. Recupere uma cópia de leitura do arquivo "*aulao403e.c*" e uma cópia de escrita dos arquivos "**makefile*".
66. Limpe o diretório (*make clean-all*).