

# Aula Prática 06 - Roteiro

## 14/05/2015 - Roteiro referente a aula prática 06 - Geração e Validação de Números PIS/PASEP.

Todos os arquivos solicitados a seguir deverão ser criados no diretório "Aulas-Praticas" (ver aula 1).

O PIS/PASEP é um identificador composto por dez dígitos mais um dígito verificador. Este dígito verificador é calculado através da seguinte regra:

- Multiplique os números, da esquerda para a direita, respectivamente por 3, 2, 9, 8, 7, 6, 5, 4, 3 e 2.
- Some os resultados das multiplicações.
- Calcule o resto da divisão desta soma por 11.
- Se o resto for igual a 0 ou igual a 1 o dígito verificador será igual a 0.
- Caso contrário, o dígito verificador será igual à diferença entre 11 e o resto calculado acima.

Como exemplo, para o número 1701209041-1, o cálculo seria:

$$1 \times 3 + 7 \times 2 + 0 \times 9 + 1 \times 8 + 2 \times 7 + 0 \times 6 + 9 \times 5 + 0 \times 4 + 4 \times 3 + 1 \times 2 = 98$$

Como o resto da divisão de 98 por 11 é 10 e como a diferença entre 11 e este resto é 1, o dígito verificador será igual ao resto, ou seja, 1.

1. Baseado nas definições acima, crie o arquivo "aula0601.h" contendo os protótipos definidos abaixo. Este arquivo deverá conter também as macros e os tipos necessários para a implementação destas funções. A macro referente à combinação *ifndef* e *define*, como por exemplo `_AULA0601_`, deverá ser definida como uma *string* valendo:

```
"@(#)aula0601.h $Revision$"
```

```
byte
```

```
GerarDigitoVerificadorPisPasep (byte [ ]);
```

```
boolean
```

```
ValidarPisPasep (byte [ ]);
```

A função *GerarDigitoVerificadorPisPasep* deverá receber os 10 primeiros dígitos de um número PIS/PASEP (como um vetor de bytes) e deverá retornar o dígito verificador correspondente.

A função *ValidarPisPasep* deverá receber os 11 dígitos de um número PIS/PASEP (como um vetor de bytes) e deverá retornar **verdadeiro** se o dígito verificador for válido. Caso contrário, deverá retornar **falso**. Esta função deverá utilizar a função *GerarDigitoVerificadorPisPasep*.

2. Crie o arquivo "aula0601.c" contendo a implementação da função *GerarDigitoVerificadorPisPasep*.
3. Crie o arquivo "aula0602a.c" contendo a implementação de um programa de testes para a função *GerarDigitoVerificadorPisPasep*. Este programa deverá receber os 10 primeiros dígitos do número PIS/PASEP desejado através de 10 argumentos da linha de comando (CLI). O programa deverá exibir o dígito verificador correspondente. Todos os tratamentos de erro necessários e que não possam ser realizados na função *GerarDigitoVerificadorPisPasep* deverão ser implementados neste programa.
4. Inclua, no arquivo de dependências, a macro `AULA0602Aobjs` - correspondendo aos

arquivos necessários para criar o executável a partir dos arquivos "aula0601.c" e "aula0602a.c". Além disso, defina a macro AULA06, equivalendo ao executável "aula0602a", e altere a macro EXECS, de forma que o valor da mesma inclua os executáveis criados na aula 06. O arquivo de dependências deverá incluir ainda os objetivos *aula06* (todos os executáveis da aula 06) e *aula0602a* (executável criado a partir dos arquivos definidos pela macro AULA0602AOBJS) com os comandos correspondentes.

5. Compile, linkedite e teste o programa na máquina local (*CentOS*) e na *loghost02 (FreeBSD)*.
6. Submeta os arquivos "aula0601.h", "aula0601.c", "aula0602a.c" e "Makefile" ao sistema de controle de versão *RCS* (comando *ci*).
7. Recupere uma cópia de leitura dos arquivos "aula0602a.c" (comando *co*) e uma cópia de escrita dos arquivos *aula0601.h*, "aula0601.c" e "Makefile" (comando *co -l*).
8. Crie o arquivo "aula0602b.c" contendo a implementação de um programa de testes para a função *GerarDigitoVerificadorPisPasep*. Este programa deverá receber os 10 primeiros dígitos do número PIS/PASEP desejado através de 1 argumento da linha de comando (CLI). O programa deverá exibir o dígito verificador correspondente. Todos os tratamentos de erro necessários e que não possam ser realizados na função *GerarDigitoVerificadorPisPasep* deverão ser implementados neste programa.
9. Inclua, no arquivo de dependências, a macro AULA0602BOBJS - correspondendo aos arquivos necessários para criar o executável a partir dos arquivos "aula0601.c" e "aula0602b.c". Além disso, altere a macro AULA06, de forma que o valor da mesma inclua o executável "aula0602b". O arquivo de dependências deverá incluir ainda o objetivo *aula0602b* (executável criado a partir dos arquivos definidos pela macro AULA0602BOBJS) com os comandos correspondentes.
10. Compile, linkedite e teste o programa na máquina local (*CentOS*) e na *loghost02 (FreeBSD)*.
11. Submeta os arquivos "aula0602b.c" e "Makefile" ao sistema de controle de versão *RCS* (comando *ci*).
12. Recupere uma cópia de leitura do arquivo "aula0602b.c" e uma cópia de escrita do arquivo "Makefile" (comando *co -l*).
13. Inclua, no arquivo "aula0601.c", a implementação da função *ValidarPisPasep*.
14. Crie o arquivo "aula0603a.c" contendo a implementação de um programa de testes para a função *ValidarPisPasep*. Este programa deverá receber os 11 dígitos do PIS/PASEP desejado através de 11 argumentos da linha de comando (CLI) e deverá exibir se o PIS/PASEP em questão é válido ou inválido. Todos os tratamentos de erro necessários e que não possam ser realizados na função *ValidarPisPasep* deverão ser implementados neste programa.
15. Inclua, no arquivo de dependências, a macro AULA0603AOBJS - correspondendo aos arquivos necessários para criar o executável a partir dos arquivos "aula0601.c" e "aula0603a.c". Além disso, altere a macro AULA06, de forma que o valor da mesma inclua o executável "aula0603a". O arquivo de dependências deverá incluir ainda o objetivo *aula0603a* (executável criado a partir dos arquivos definidos pela macro AULA0603AOBJS) com os comandos correspondentes.
16. Compile, linkedite e teste o programa na máquina local (*CentOS*) e na *loghost02 (FreeBSD)*.
17. Submeta os arquivos "aula0601.c", "aula0603a.c" e "Makefile" ao sistema de controle de versão *RCS* (comando *ci*).
18. Recupere uma cópia de leitura dos arquivos "aula0601.c" e "aula0603a.c" e uma cópia de escrita do arquivo "Makefile" (comando *co -l*).
19. Crie o arquivo "aula0603b.c" contendo a implementação de um programa de testes para a função *ValidarPisPasep*. Este programa deverá receber os 11 dígitos do PIS/PASEP desejado através de 1 argumento da linha de comando (CLI) e deverá exibir se o PIS/PASEP em questão é válido ou inválido. O argumento recebido deverá incluir o hífen. Todos os tratamentos de erro necessários e que não possam ser realizados na função *ValidarPisPasep* deverão ser implementados neste programa.
20. Inclua, no arquivo de dependências, a macro AULA0603BOBJS - correspondendo aos arquivos necessários para criar o executável a partir dos arquivos "aula0601.c" e "aula0603b.c". Além disso, altere a macro AULA06, de forma que o valor da mesma inclua o executável "aula0603b". O arquivo de dependências deverá incluir ainda o objetivo *aula0603b* (executável criado a partir dos arquivos definidos pela macro AULA0603BOBJS) com os comandos correspondentes.
21. Compile, linkedite e teste o programa na máquina local (*CentOS*) e na *loghost02 (FreeBSD)*.
22. Submeta os arquivos "aula0603b.c" e "Makefile" ao sistema de controle de versão *RCS* (comando *ci*).
23. Recupere uma cópia de leitura do arquivo "aula0603b.c" e cópia de escrita do arquivo

"*Makefile*" (comando *co -l*).

24. Repita todos os itens anteriores trocando o vetor de bytes por uma *string* e renumerando corretamente os arquivos, ou seja:
- "*aulao301.h*" será renomeado para "*aulao304.h*".
  - "*aulao301.c*" será renomeado para "*aulao304.c*".
  - "*aulao302a.c*" será renomeado para "*aulao305a.c*".
  - "*aulao302b.c*" será renomeado para "*aulao305b.c*".
  - "*aulao303a.c*" será renomeado para "*aulao306a.c*".
  - "*aulao303b.c*" será renomeado para "*aulao306b.c*".

char

GerarDigitoVerificadorPisPasep (char [ ]);

boolean

ValidarPisPasep (char [ ]);

A função *ValidarPisPasep* deverá receber uma *string* que inclua o hífen.

Observações:

- Todos os arquivos criados deverão incluir comentários com as seguintes informações:  
Universidade Federal do Rio de Janeiro  
Escola Politecnica  
Departamento de Eletronica e de Computacao  
EEL270 - Computacao II - Turma 2015/1  
Prof. Marcelo Luiz Drumond Lanza  
Autor: Nome Completo  
  
\$Author\$  
\$Date\$  
\$Log\$
- No final do arquivo inclua um comentário com a macro \$RCSfile\$.
- Prazo: 21/05/2015 (início da aula prática 07).