

Trabalho Prático - Simulador CESAR

Escrever um programa para o simulador **Cesar** que deve identificar e validar um número de cartão de crédito. Para isso, utilizar a metodologia descrita a seguir.

Identificação da companhia do cartão

Um cartão é composto de uma seqüência de 13 a 16 dígitos decimais. Os dígitos mais significativos (prefixos) identificam qual a companhia que opera o cartão, de acordo com a seguinte tabela:

Operadora	Prefixo	Tamanho (em dígitos)	Algoritmo de Verificação
Mastercard	51 a 55	16	Luhn 10
Visa	4	13 ou 16	Luhn 10
Amex	34 ou 37	15	Luhn 10
Diners	30 ou 36 ou 38	14	Luhn 10
Discover	6011	16	Luhn 10
enRoute	2014 ou 2149	15	Luhn 10
JCB	3	16	Luhn 10
JCB	2131 ou 1800	15	Luhn 10

Algoritmo de Luhn

O algoritmo de Luhn foi desenvolvido por Hans Peter Luhn, da IBM, em 1954. Ele é capaz de detectar um erro simples em um dígito, assim como a maioria das transposições de um dígito com os dígitos adjacentes. Este algoritmo verifica se um número de cartão é bem formado realizando o seguinte teste:

1. Iniciando com o dígito mais à direita, multiplique por 2 (dois) todos os dígitos de posição par, e deixe inalterados os dígitos das posições ímpares. Note que o dígito mais à direita é indexado por 1, ou seja, sua posição sempre é ímpar. Por exemplo, o número 49927398716 seria reescrito como 4(18)9(4)7(6)9(16)7(2)6. Os parênteses ilustram os dígitos que foram dobrados, e são usados apenas para fins didáticos. Sem os parênteses, o número resultante seria 4189476916726.
2. Some todos os dígitos, independente da sua posição. Assim, o número resultante do passo 1 produziria $4+1+8+9+4+7+6+9+1+6+7+2+6 = 70$
3. Se o número final é congruente a zero em módulo 10 (ou, em termos mais simples, se a divisão por dez produzir um resto igual a zero), então o número inicial é válido. Note que, obviamente, isso não quer dizer que o cartão seja legítimo, ou que tenha crédito, ou que não tenha expirado. Passar no teste de Luhn quer simplesmente dizer que não ocorreram erros, ou seja, o número é bem formado.

Especificação do programa

O programa deve iniciar identificando o autor, listando no visor o seu nome e o seu número de identificação UFRGS. Caso sejam utilizadas várias linhas, deve-se aguardar que seja digitada a tecla *Enter* (código ASCII 0x0D) após cada linha, antes de passar para a próxima.

Após esta identificação, escrever nas posições mais a esquerda do visor os caracteres “#>”, e aguardar a digitação de um número de cartão. Cada caractere digitado deve ser ecoado no visor. Além disso, deve ser apresentado (e controlado) um cursor, representado pelo caractere *Underscore* (“_”, código ASCII 0x5F). Também deve ser permitido o uso da tecla *Backspace* (código ASCII 0x08) para corrigir erros de digitação (Cuidado com o tratamento do *Backspace* antes do primeiro caractere!). Um número termina quando for digitada a tecla *Enter*, ou quando for atingido o final do visor.

Quando terminar a digitação de um número, o programa deve identificar a operadora, de acordo com a tabela fornecida anteriormente, e verificar a validade do número. Note que um número será válido se passar no teste de Luhn e tiver o tamanho (número de dígitos) correto.

As situações a serem tratadas e que devem ser indicadas por mensagens adequadas são as seguintes (a definição exata da mensagem fica a seu critério, mas não deve ultrapassar uma linha do visor):

1. Números que contiverem caracteres não numéricos (caracteres fora do intervalo [0;9]);
2. Números com tamanho incorreto;
3. Números de operadoras desconhecidas (que não precisam ser validados);
4. Números de operadoras conhecidas, mas inválidos;
5. Números de operadoras conhecidas e válidos.

Nos casos (4) e (5), a mensagem deve incluir o nome da operadora.

Após processar o número e escrever no visor a mensagem adequada, o programa deve aguardar que seja digitado um *Enter* e então escrever no visor “Validar outro ou Terminar?”. Se for digitado “t” ou “T”, o programa deve finalizar, escrevendo uma eventual mensagem de encerramento. Se for digitado “v” ou “V”, o programa deve tratar outro número, reiniciando a partir do ponto em que é escrito “#>” no visor.

Não esqueça que os caracteres digitados pelo usuário são recebidos em ASCII e podem ser enviados para o visor neste formato. No entanto, para realizar o teste de Luhn é necessário que os dígitos 0 a 9 (códigos ASCII de 0x30 a 0x39) sejam convertidos para valores numéricos.

Na avaliação do trabalho será levada em consideração a “interface com o usuário“. Quanto mais amigável e intuitiva for, melhor poderá ser a nota obtida.

O trabalho deverá ser feito de forma individual e entregue via Moodle, contendo um arquivo fonte comentado (.ced ou .txt) e um executável (.mem). Para nomear os arquivos, utilize os seus dois primeiros nomes, sem espaço em branco e sem acentos. Por exemplo, o aluno **Um Três Dois de Oliveira Quatro** deverá nomear seus arquivos **UmTres.ced** e **UmTres.mem**. Os arquivos do trabalho deverão ser compactados em um arquivo **UmTres.rar** (ou **UmTres.zip**) e entregues através do Moodle.

Data final de entrega: dia 30 de setembro, conforme especificado no Moodle

Os testes serão realizados com a opção “Atualizar Registradores” desligada (no Menu “Executar”). Leve isto em conta caso o seu programa utilize rotinas que dependam de tempo (como um cursor piscante, por exemplo).

Casos de teste

Número	Caso
354683A6c	Caracteres inválidos
347598	Número muito curto
51783746281726178	Número muito longo
7145819143526709	Operadora desconhecida
355601236748643	Operadora desconhecida
213078347298518	Operadora desconhecida
5105105105105100	Mastercard, válido
5555555555554443	Mastercard, inválido
422222222222	Visa, válido
4567890123456789	Visa, inválido
378282246310005	Amex, válido
38520000023237	Diners, válido
3566002020360606	JCB, inválido