

Lista de Exercícios N° 1

Esta lista pode ser desenvolvida em grupos de até dois alunos, seguindo as especificações contidas no arquivo 00_ProcListas.pdf disponível na área pública da disciplina na rede. A entrega ao professor deverá ocorrer até o dia 22 de maio de 2018.

1) O dia da semana para uma data qualquer pode ser calculado pela seguinte fórmula:

$$\text{Dia da semana} = \text{RESTO}(\text{TRUNCA}(2.6 * M - 0.1) + D + A + \text{QUOCIENTE}(A, 4) + \text{QUOCIENTE}(S, 4) - 2 * S, 7)$$

onde:

- M representa o número do mês. Janeiro e fevereiro são os meses 11 e 12 do ano precedente, março é o mês 1 e dezembro é o mês 10.
- D representa o dia do mês.
- A representa o número formado pelos dois últimos algarismos do ano.
- S representa o número formado pelos dois primeiros algarismos do ano.

Os dias da semana são numerados de zero a seis; domingo corresponde a 0, segunda a 1, e assim por diante. Faça então um programa que:

- Leia um conjunto de datas (dia, mês, ano), conforme os exemplos apresentados a seguir. Assuma que todas as datas informadas serão datas válidas. Encerrar o processamento quando for informado a data dia 0 mês 0 ano 0, que não deverá ser processada.
- Para cada data lida, determine o dia da semana correspondente, segundo o método especificado anteriormente. Caso a aplicação da fórmula produza um valor negativo, some 7 a esse valor e prossiga normalmente com o processamento.
- Escreva, para cada data lida, o dia, o mês, o ano e o nome do dia da semana calculado pelo seu programa, no formato indicado nos exemplos.

Dicas:

- O resto da divisão do número inteiro x pelo número inteiro y é um inteiro dado, em linguagem C, por $x \% y$.
- O quociente da divisão do número inteiro x pelo número inteiro y é um inteiro dado, em linguagem C, por x / y . Se um dos operandos (x ou y ou ambos) for de tipo real (float ou double), então o resultado será um número real.
- Para truncar um número real, descartando sua parte fracionária e convertendo o resultado para inteiro, basta colocar a expressão (int) antes do número ou variável real. Por exemplo, (int)3.1415 produz o valor 3; (int)vInfo produz o valor 10, caso a variável real vInfo possua o valor 10.424397.

Exemplos:

Valor informado	Resultado a ser exibido
31 3 1991	31/03/1991: Domingo
5 12 1984	05/12/1984: Quarta-Feira
8 5 1998	08/05/1998: Sexta-Feira

Lista de Exercícios N° 1

- 2) Faça um programa para validar números de CPF (antigo CIC) conforme as regras descritas ao final deste enunciado. Para cada CPF informado o programa deverá emitir a mensagem 'válido' ou então 'inválido – esperado: 99, encontrado: 99', onde o primeiro 99 indica o valor correto dos dígitos verificadores e o segundo 99 representa o par de dígitos verificadores efetivamente encontrado no CPF informado. Considere que todo CPF a ser processado será informado exatamente na forma '999.999.999-99'. Encerrar o processamento quando o usuário informar o CPF '000.000.000-00'.

Visão geral:

O CPF é composto por 11 dígitos numéricos, sendo que os 2 últimos correspondem aos chamados dígitos verificadores, que são calculados com base nos valores e na posição dos 9 dígitos iniciais.

1	2	3		4	5	6		7	8	9		d1	d2
9	9	9	.	9	9	9	.	9	9	9	-	9	9

Inicialmente se calcula o valor do primeiro dígito, que será utilizado posteriormente para o cálculo do segundo dígito.

Forma do cálculo:

- Um CPF é completado com zeros à esquerda até atingir o comprimento padrão.
- Se todos os dígitos numéricos do CPF informado forem iguais (por exemplo “222.222.222-22”) o CPF é inválido.
- Multiplicar cada um dos 9 dígitos iniciais pelo número correspondente a sua posição no CPF, e somar os resultados de cada multiplicação:

$$(\text{Num1} * 1) + (\text{Num2} * 2) + (\text{Num3} * 3) + \dots + (\text{Num9} * 9)$$

- Calcular o resto da divisão do valor obtido na soma por 11. Esse resto é o primeiro dígito verificador e, caso seja igual a 10, considerá-lo como sendo o número zero.
- Multiplicar cada um dos 10 dígitos disponíveis (os 9 do CPF mais o primeiro dígito verificador já obtido) por um valor próprio, conforme a seguinte regra: o primeiro (o mais à esquerda) por 11, o segundo por 10, o terceiro por 9, sucessivamente até o último (o mais à direita, que é o primeiro dígito verificador correto) que será multiplicado por 2, e somar os resultados de cada multiplicação:

$$(\text{Num1} * 11) + (\text{Num2} * 10) + (\text{Num3} * 9) + \dots + (\text{Num10} * 2)$$

- Multiplicar o valor obtido com a soma por 10 e, em seguida, calcular o resto da divisão desse resultado por 11. Esse resto é o segundo dígito verificador e, caso seja igual a 10, considerá-lo como sendo o número zero.

Exemplos de CPFs válidos:

823.829.839-30

282.121.299-26

214.244.321-46

123.434.839-06

321.321.325-00

Lista de Exercícios N° 1

- 3) O Centro Paula Souza vem passando por significativas modificações nos últimos anos e uma deles se refere ao software para controle acadêmico que foi implantado de forma unificada para todas as Fatecs. O SIGA (esse é o nome do sistema de gestão) padronizou o registro acadêmico em todas as unidades, além de oferecer uma série de recursos muito interessantes para os professores, coordenadores de curso e secretários acadêmicos. Uma das mudanças mais perceptíveis proporcionadas pela implantação do SIGA foi a mudança no formato dos números de matrícula dos alunos, os chamados RAs. Antigamente esse dado variava grandemente entre as unidades, agora todas usam o mesmo formato, de 13 dígitos numéricos, na forma FFFCCCAASTNNN, onde:

FFF = Código numérico da Fatec (Sorocaba é 003, São Paulo é 002, etc)

CCC = Código do curso

048 – Análise e Desenvolvimento de Sistemas

061 – Sistemas Biomédicos

073 – Eletrônica Automotiva

074 – Logística

080 – Polímeros

081 – Processos Metalúrgicos

099 – Projetos Mecânicos

100 – Fabricação Mecânica

AA = Ano de ingresso do aluno

S = Semestre de ingresso do aluno (1=primeiro semestre, 2= segundo semestre)

T = Turno do curso (1=manhã/matutino, 2=tarde/vespertino, 3=noite/noturno)

NNN = Número sequencial do aluno

Por exemplo, o novo RA 0030481321099 significa

Fatec:	003 (Sorocaba)
Curso:	048 (Análise e Desenvolvimento de Sistemas)
Ano de ingresso:	13 (2013)
Semestre de ingresso:	2 (Segundo semestre)
Turno:	1 (manhã)
Número sequencial:	099

Na Fatec de Sorocaba, por exemplo, o RA antigo era composto por 8 caracteres, na forma CCAASNNN, onde:

CC = Sigla do curso

AD – Análise e Desenvolvimento de Sistemas (manhã)

AN – Análise e Desenvolvimento de Sistemas (noite)

SD – Sistemas Biomédicos (manhã)

LT – Logística (tarde)

PL – Polímeros (tarde)

PD – Projetos Mecânicos (manhã)

PN – Projetos Mecânicos (noite)

OD – Fabricação Mecânica (manhã)

ON – Fabricação Mecânica (noite)

Lista de Exercícios N° 1

AA = Ano de ingresso do aluno

S = Semestre de ingresso do aluno (1=primeiro semestre, 2= segundo semestre)

NNN = Número sequencial do aluno

Ainda temos, em Sorocaba, alunos remanescentes do período em que no ato da matrícula o RA era gerado no formato antigo. Sua equipe foi designada para gerar um programa que, para um RA antigo da Fatec de Sorocaba informado, gere uma versão correspondente no novo formato de 13 caracteres, aproveitando nos últimos três dígitos a informação correspondente do RA antigo. Por exemplo, o RA antigo AD132099 seria o equivalente ao RA novo 0030481321099.

Entrada

Inicialmente um valor N é informado, indicando a quantidade de casos de teste a serem processados. Seguem-se N linhas, cada uma contendo uma *string* de 8 caracteres representando um RA antigo válido da Fatec de Sorocaba. Não serão fornecidos RAs antigos dos cursos de Eletrônica Automotiva e de Processos Metalúrgicos pois são cursos muito recentes e já foram implantados com os novos RAs.

Saída

Para cada caso de teste imprima o novo RA de 13 caracteres correspondente ao RA antigo lido.

Exemplo

Entrada:	Saída:
9	0030481011001
AD101001	0030481223035
AN122035	0030611111111
SD111111	0030741212221
LT121221	0030800912252
PL091252	0030990821124
PD082124	0030990713255
PN071255	0031001211022
OD121022	0031001213212
ON121212	

4) Nomes abstratos (*Problema 11713 do UVa*)

Alguns de vocês devem ter observado que em determinados jogos de computadores, particularmente naqueles baseados em esportes, a grafia de nomes é modificada para que não fiquem exatamente iguais aos dos objetos e pessoas do mundo real. Isso é feito para evitar a necessidade de obter a permissão de cada jogador para usá-lo no jogo, bem como para se resguardar em algumas questões relacionadas a patentes. Neste problema você receberá um par de nomes, um dos quais é o de algum jogador do mundo real e o segundo foi encontrado em um jogo. Você deverá determinar se os dois nomes são o mesmo, isto é, se o segundo pode ser obtido por alguma modificação sobre o primeiro nome.

Dois nomes são considerados o mesmo se eles possuem o mesmo comprimento e as únicas diferenças são aquelas encontradas nas posições das vogais. Isso significa que um nome que pode ser obtido a partir de outro nome, substituindo zero ou mais vogais por outras vogais, é considerado o mesmo que o nome original, desde que possuam o mesmo comprimento. Por exemplo, tanto *polo* como *pola* são o mesmo, mas *pelet* e *bele* não.

Lista de Exercícios N° 1

Entrada

A primeira linha da entrada contém um inteiro positivo $n \leq 20$, onde n denota o número de casos de teste. Ele será seguido por $2*n$ linhas, em que cada linha conterà um nome de até 20 caracteres de comprimento. Os nomes consistirão apenas de palavras em minúsculas.

Saída

Para cada caso de teste, uma linha deverá ser impressa na saída. Ela deverá ser *Yes* se o segundo nome puder ser obtido por alguma mudança nas vogais do primeiro nome, senão deverá ser impresso *No*.

Exemplo de entrada	Saída para o exemplo de entrada
5	Yes
pele	Yes
polo	No
pele	No
pola	No
ronaldo	
ronaldino	
pele	
pelet	
pele	
bele	