

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 pasta de arquivos do canal Geral da equipe, no Teams. A entrega ao professor deverá ocorrer até o dia 24 de outubro de 2022.

1) 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.

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 conterá 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	

Lista de Exercícios N° 1

- 2) Faça um programa que recebe vários pares de pontos e, para cada par, determinar a distância entre eles, conforme a fórmula

$$\text{Distância} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Encerrar o processamento quando for recebido um conjunto de entradas com quatro números zero. Usar uma função para calcular a distância.

Exemplos

Entrada:

```
1.0  5.0  3.0  4.2
1.0  5.0  3.0  1.5
1.0  5.0  6.4  2.45
3.0  1.5  7.0  2.2
0.0  0.0  0.0  0.0
```

Saída esperada:

```
2.154066
4.031129
5.971809
4.060788
```

- 3) Escreva um programa em linguagem C que recebe os dados fornecidos pelo usuário e os armazena nas matrizes PecaMes e GranaPeca indicadas abaixo. A primeira indica a produção das peças P1, P2 e P3 nos meses de Janeiro a Março e a segunda indica o custo de cada tipo de peça bem como a respectiva margem de lucro. O programa deve então gerar e exibir na tela a matriz GranaMes, que indica os valores de custo e lucro verificados de Janeiro a Março.

<u>PecaMes</u>				<u>GranaPeca</u>		
	P1	P2	P3		CUSTO	LUCRO
JAN	1	2	3	P1	1	0
FEV	4	5	6	P2	2	1
MAR	7	8	9	P3	4	2

<u>GranaMes</u>		
	CUSTO	LUCRO
JAN	17	8
FEV
MAR

- 4) 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 e que você deve utilizar uma struct para armazenar as datas. Encerrar o processamento quando for informado a data dia 0 mês 0 ano 0, que não deverá ser processada.

Lista de Exercícios N° 1

- Para cada data lida, determine, por meio de uma função, 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

- 5) 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:

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

Lista de Exercícios N° 1

- d)** 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.
- e)** 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:
- $$(\text{Num1} * 11) + (\text{Num2} * 10) + (\text{Num3} * 9) + \dots + (\text{Num10} * 2)$$
- f)** 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