

Valor: 4,0 pontos

Correção: a questão está dividida em casos triviais, completos, visíveis e ocultos. O aluno ganhará os pontos conforme a tabela abaixo. Para receber o ponto das funções, o aluno deverá passar pelo menos um dos casos da tabela.

Casos	Pontos
Visíveis triviais	0,5
Ocultos triviais	0,5
Visíveis completos	1,0
Ocultos completos	1,0
Implementação das funções	1,0

(BOCA: P1_2024_Q3) Implemente um programa em C para validar uma lista de números de telefone fixo e/ou celular. Para um número de celular ser válido, ele deve conter exatamente 11 dígitos numéricos, incluindo o código de área (DDD). Além disso, o número imediatamente após o DDD deve ser o número 9. Já para um número de telefone fixo, ele deve conter 10 dígitos numéricos, também incluindo o DDD, mas não pode conter o número 9 após o código de área. Em ambos os casos, o DDD deve pertencer ao conjunto de códigos de área válidos. Além disso, nesse exercício, um número pode aparecer de dois jeitos: com parênteses e um traço no meio do número ou apenas os números. Por exemplo: (27)91111-1111 ou 27911111111.

Para esse exercício, os seguintes DDDs são considerados válidos:

- 27: VITORIA
 - 41: CURITIBA
 - 61: BRASILIA
 - 81: RECIFE
 - 92: MANAUS
-
- Entrada: A primeira linha da entrada contém um número inteiro N, que representa a quantidade de números de telefone que serão fornecidos. As N linhas seguintes contém, cada uma, um número de telefone com todos os dígitos em sequência sem espaços. Além disso,

cada linha termina em “\n”. Nos casos triviais todos os números aparecerão com parênteses e um traço no meio do número.

- Saída: Para cada número analisado, o programa deve imprimir uma mensagem indicando se o número possui um DDD e se é um tipo de número válido ou inválido. A mensagem de saída deve seguir o seguinte padrão “DDD X TIPO Y”, onde X pode ser INVALIDO ou a cidade correspondente do DDD válido (VITORIA, CURITIBA, BRASILIA, RECIFE, MANAUS). Já Y pode ser apenas CELULAR, FIXO ou INVALIDO.

O programa deve estar modularizado, para isso crie:

- a) (0.25 pts) Uma função responsável por coordenar a chamada das demais funções de validação e impressão. Cabeçalho: `void ValidarNumero();`
- b) (0.25 pts) Uma função que leia o código de área e verifique se ele é válido. Além disso, retorne o código de área caso seja válido ou 0 se for inválido. Considere que a função será chamada no início da linha, ou seja, a próxima entrada corresponde ao início do número antes do DDD. Além disso, a função deve consumir as entradas até o final do DDD. Cabeçalho: `int LeCodigo();`
- c) (0.25 pts) Uma função que consuma os demais dígitos após o DDD para determinar se é um celular ou telefone fixo, retornando respectivamente 1 ou 2. Caso seja inválido retorne 0. A função deve assumir que o DDD já foi consumido, ou seja, a leitura começa a partir do primeiro dígito após o DDD. Além disso, essa função também é responsável por consumir o restante da linha até o caractere “\n”. Cabeçalho: `int LeTipo();`
- d) (0.25 pts) Uma função que recebe como parâmetros os retornos das funções das letras b e c para imprimir a mensagem apropriada conforme as regras de saída. Cabeçalho: `void ImprimirInformacoes(int ddd, int tipoNum);`