

Instruções gerais para o BOCA: O BOCA é um programa de correção automática dos exercícios. Portanto, é necessário seguir estritamente os padrões de entrada e saída das questões. Apesar de, em um programa comum para um usuário qualquer, você ter que usar `printf's` para pedir uma entrada de dados para o usuário, em programas para o BOCA os `printf's` e `scanf's` devem ser usados com muita cautela, e seguindo rigorosamente os formatos de entrada e saída definidos para cada questão a ser submetida para o BOCA. O sistema compara letrinha por letrinha da saída do seu programa (isto é, tudo que foi escrito na tela) com a saída esperada por ele, portanto tudo deve ser escrito na saída padrão (ex. tela do monitor) conforme indicado nos exemplos das questões. Qualquer `printf` realizado sem necessidade pode invalidar a resposta. Lembre-se que os exemplos dados podem não cobrir todos os casos de teste das questões.

(BOCA: P1_2019_Q4) Problema: Faça um programa que verifique a validade de um CPF. Um CPF é composto por 11 dígitos numéricos `#1#2#3.#4#5#6.#7#8#9-#10#11`, sendo os nove primeiros a base e os dois últimos os verificadores. Para que um CPF seja válido, seus dígitos verificadores devem obedecer a um cálculo feito a partir dos outros dígitos do CPF. Um CPF com todos dígitos iguais já é inválido independente do cálculo do dígito verificador.

Cálculo do dígito verificador #10:

$soma = \#1 * 10 + \#2 * 9 + \#3 * 8 + \#4 * 7 + \#5 * 6 + \#6 * 5 + \#7 * 4 + \#8 * 3 + \#9 * 2$

Se o resto da divisão da soma por 11 for menor que 2, então o dígito é 0

Se o resto da divisão da soma por 11 for maior ou igual 2, então o dígito é (11 - resto)

Cálculo do dígito verificador #11:

$soma = \#1 * 11 + \#2 * 10 + \#3 * 9 + \#4 * 8 + \#5 * 7 + \#6 * 6 + \#7 * 5 + \#8 * 4 + \#9 * 3 + \#10 * 2$

Se o resto da divisão da soma por 11 for menor que 2, então o dígito é 0

Se o resto da divisão da soma por 11 for maior ou igual 2, então o dígito é (11 - resto)

- Entrada: Um CPF no formato `###.###.###-##`. Veja o formato de entrada nos exemplos abaixo.
- Saída: A saída deverá imprimir: "CPF valido!" se o CPF informado for válido; "CPF invalido: digitos iguais!" se ele for inválido por ter todos os dígitos iguais; "CPF invalido: primeiro digito!" se ele for inválido por ter o primeiro dígito verificador diferente do esperado; "CPF invalido: segundo digito!" se ele for inválido por ter o segundo dígito verificador diferente do esperado; e, "CPF invalido: dois digitos!" se ele for inválido por ter os dois dígitos verificadores diferentes dos esperados. Veja o formato da saída nos exemplos abaixo.

- Exemplo de Entrada:
999.999.999-99
111.444.777-35
111.444.777-25
111.444.777-34
111.444.777-23
529.982.247-25
- Exemplo de Saída:
CPF invalido: digitos iguais!
CPF valido!
CPF invalido: primeiro digito!
CPF invalido: segundo digito!
CPF invalido: dois digitos!
CPF valido!