

## Problema: Validador de CPF

Arquivo: `cpfvalido.c`

Cadastro de Pessoa Física (CPF) é um documento brasileiro que é emitido pela Secretaria da Receita Federal do Ministério da Fazenda. É composto de um número de 11 dígitos, sendo os dois últimos os dígitos os verificadores (dv). Os oito primeiros dígitos designados aleatoriamente no momento da inscrição, o 9º dígito indica a região fiscal de onde o CPF foi emitido e o 10º e o 11º dígitos validam o documento como um todo, utilizando o algoritmo Módulo 11.



O algoritmo do Módulo 11 soma o produto de cada dígito do CPF por um peso e calcula o resto da divisão por 11. Por 11 ser um número primo, os dígitos produzidos por este algoritmo tem a capacidade de identificar tanto erros na substituição de dígitos, quanto na transposição deles. Para contornar o resto da divisão por 11 que pode resultar em 10, a Receita Federal estabeleceu que o este 10 gerado fosse substituído por 0 e, para minimizar erro de verificação, introduziu o segundo dígito verificador que também é calculado pelo mesmo algoritmo, mas incluindo em seu cálculo o 1º dígito verificador calculado.

Por exemplo, considere que gostaríamos de calcular os dígitos verificadores do CPF 123.456.789-XX.

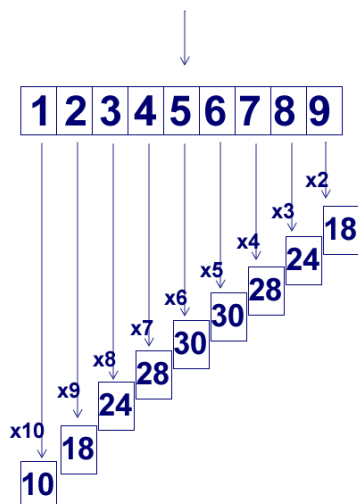
# 123.456.789-XX



1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

Do dígito mais significativo para o menos significativo, distribuiremos os pesos 10, 9, 8, 7, 6, 5, 4, 3 e 2. Após o multiplicar cada dígito por seu respectivo peso, calcularemos o somatório deles.

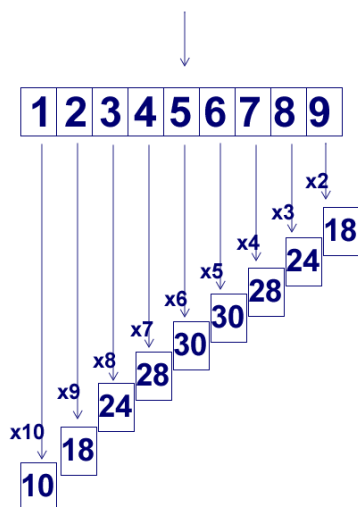
**123.456.789-XX**



$$10+18+24+28+30+30+28+24+18 = 210$$

Com resultado desta soma, nós calcularemos o resto da divisão dela por 11. Após isso, faremos a subtração de 11 pelo resto para calcular o 1º dígito verificador. Se este resultado for maior que 9, substituiremos esse dígito por 0.

**123.456.789-XX**



$$10+18+24+28+30+30+28+24+18 = 210$$

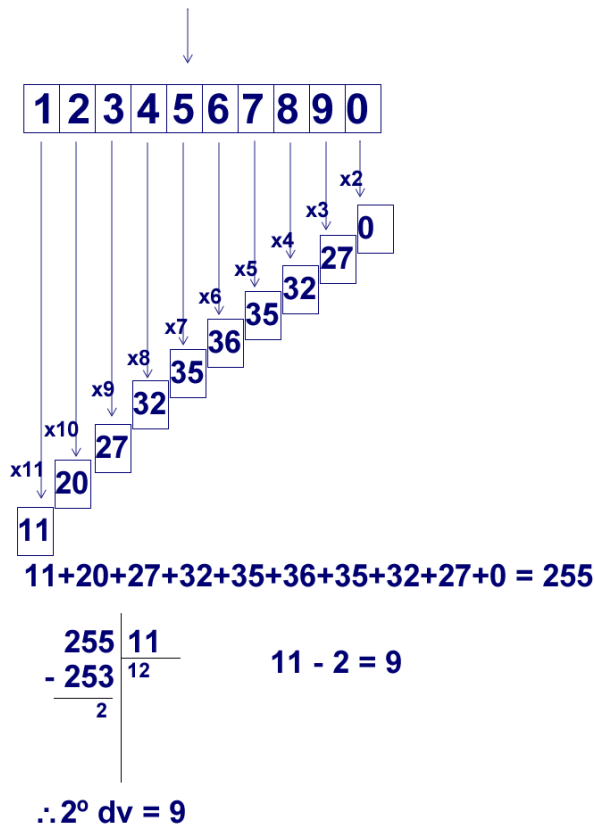
$$\begin{array}{r|l} 210 & 11 \\ - 209 & 19 \\ \hline & 1 \end{array}$$

$$11 - 1 = 10$$

$$\therefore 1^\circ \text{ dv} = 0$$

Faremos a mesma operação para o cálculo do segundo dígito, incluindo em sua conta o 1º dígito verificador.

**123.456.789-0X**



Desta forma, o CPF completo incluindo os dígitos verificadores será 123.456.789-09.

Escreva um programa em C que recebe os 11 dígitos do CPF do usuário e exiba se é um CPF válido ou não.

---

*Cadastro de Pessoa Física (CPF)* is a brazilian document issued by *Secretaria da Receita Federal do Ministério da Fazenda*. It is 11-digit number where the last two digits are called *dígitos verificadores (dv, or verification digits)*. The first 8 numbers are randomly generated, the 9th digit indicates the region where this CPF comes from and the 10th and 11th validate the document as a whole, using Módulo 11 algorithm.

The Módulo 11 algorithm sums the product of each CPF digit by a factor and calculates its remainder by 11. Since 11 is a prime number, this generated number can identify if any number has been swapped or any number has been changed. Since the modulus operator by 11 can generate the number 10, the *Receita Federal* established that if this happens, this digits would be changed to 0 and, to minimize verification error, it introduced the 2th dv number which can also be calculated using Módulo 11 algorithm, including the 1th dv number in your calculation.

For example, let's calculate the dv number of the CPF 123.456.789-XX.

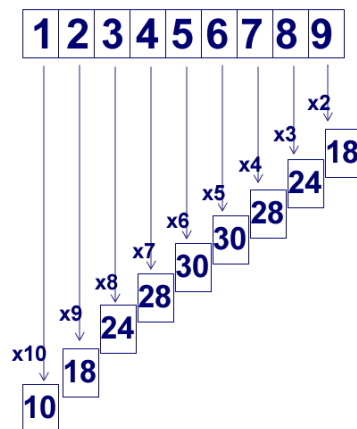
# 123.456.789-XX



1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

From the most significant digit to the least significant, let's distribute the factors 10, 9, 8, 7, 6, 5, 4, 3, 2. After multiply each digit by your respective factor, let's calculate the sum of them.

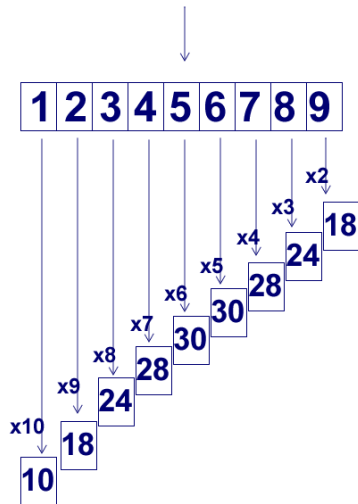
## 123.456.789-XX



$$10+18+24+28+30+30+28+24+18 = 210$$

Now, we calculate the modulus of the result of the last sum we did it. After that, we subtract by 11 with the remainder. The 1th dv digit is this result, however if this result is greater than 9, than the dv will be 0.

**123.456.789-XX**



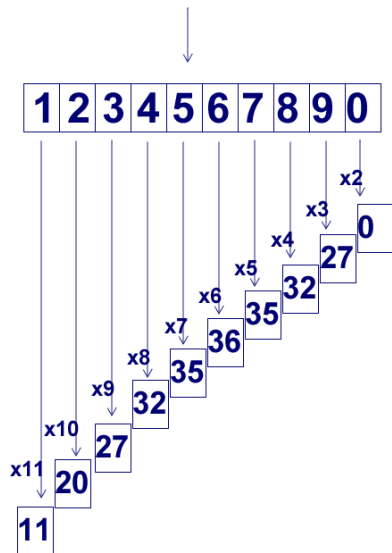
$$10+18+24+28+30+30+28+24+18 = 210$$

$$\begin{array}{r|l} 210 & 11 \\ - 209 & 19 \\ \hline & 1 \end{array} \quad 11 - 1 = 10$$

$$\therefore 1^{\circ} dv = 0$$

We will do the same thing for the second dv, including the 1th dv in its calculation.

**123.456.789-0X**



$$11+20+27+32+35+36+35+32+27+0 = 255$$

$$\begin{array}{r|l} 255 & 11 \\ - 253 & 12 \\ \hline & 2 \end{array} \quad 11 - 2 = 9$$

$$\therefore 2^{\circ} dv = 9$$

Therefore, the complete CPF will be 123.456.789-09.

Write a C program where the user inputs the 11 digits of his/her CPF and the program shows if it's valid CPF or not.

## Entrada

A primeira linha da entrada contém um número inteiro que corresponde ao número de CPF com 11 dígitos.

O número do CPF pode assumir um valor entre 00000000000 à 99999999999, ou seja:

$$00000000000 \leq \text{cpf} \leq 99999999999$$

Para lidar com este tipo de número em C, declare a variável do CPF como *unsigned long long int*, que possui o formato *%llu* tanto para leitura, quanto para escrita. Considere o seguinte trecho de código para auxílio.

```
unsigned long long int inteiroGrande;
int inteiroMenor;

scanf("%llu", &inteiroGrande); //suponha 12345678909

inteiroMenor = inteiroGrande/100;

printf("%llu\n", inteiroGrande); //vai imprimir 12345678909
printf("%d\n", inteiroMenor); //vai imprimir 123456789
```

---

The first line on the input contains one integer number which indicates the CPF number with 11 digits.

The CPF number may assume a value between 00000000000 to 99999999999, as in:

$$00000000000 \leq \text{cpf} \leq 99999999999$$

To deal with this large number in C, declare the CPF variable as *unsigned long long int*, using format *%llu* for writing and reading. Consider the following example to better understanding.

```
unsigned long long int bigInteger;
int smallInteger;

scanf("%llu", &bigInteger); //suppose 12345678909

smallInteger = bigInteger/100;

printf("%llu\n", bigInteger); //will show 12345678909
printf("%d\n", smallInteger); //will show 123456789
```

## Saída

Seu programa deve mostrar somente se o CPF é válido ou não, com a mensagem *valido* ou *invalido*. Verifique os exemplos para entender melhor o formato da saída.

---

Your program must show only if the CPF number is valid or not, with the message *valido* to indicates that is a valid CPF number or *invalido* to indicates that is not a valid CPF number.

Check the examples below for better understanding on how the output must be.

### Exemplos

Exemplo de entrada	Saída para o exemplo de entrada
12345678909	valido

Exemplo de entrada	Saída para o exemplo de entrada
89370443079	invalido