Problema: Logradouro

O formato *CSV*, **C**omma-**S**eparated-**V**alues, é um tipo de arquivo que armazena dados com valores separados por vírgula. No Brasil, assim como em outros países latinos, casas decimais de números também são separadas por *vírgula*, então, para não causar conflitos, um arquivo *CSV* criado por uma máquina que utiliza Windows localizada no Brasil utiliza o caracter *ponto-e-vírgula* para separar os dados. Cada linha do arquivo, ao abrir utilizando programas como Microsoft Excel ou Google Sheets, será transformada em uma linha na planilha, e cada coluna na planilha terá o dado que está separado por ponto-e-vírgula. Por exemplo, suponha que você esteja montando uma tabela onde a primeira coluna represente o tipo de logradouro, a segunda coluna o nome do logradouro e a terceira coluna o complemento de um endereço:

Rua	Afonso Camargo	Loja B
Avenida	da Paz	

O *CSV* correspondente seria:

```
Rua;Afonso Camargo;Loja B
Avenida;da Paz;
```

Faça uma função chamada *criaLinhaCSV*, que recebe uma estrutura do tipo *tipoLogradouro*, onde esta possui informações de tipo, nome, e complemento, e que recebe também uma string, com no máximo 240 posições, que será a linha do *CSV* gerado, onde o separador de dados deverá ser o caracter *ponto-e-vírgula*. Para este problema, não há a necessidade de realizar validações. Considere que os nomes possuem apenas letras de A à Z, maiúsculo e minúsculos, e espaços.

```
void criaLinhaCSV(struct tipoLogradouro info, char linha[240]);
```

Cada argumento da função representa, respectivamente:

- As informações do logradouro, armazenadas em uma variável do tipo *tipoLogradouro*, e;
- As informações que deverão ser salvas: O tipo, o nome e o complemento do logradouro respectivamente, da forma <**TIPO**>;<**NOME**>;<**COMPLEMENTO**>. Considere que esta string esteja vazia ao passar ela para a função.

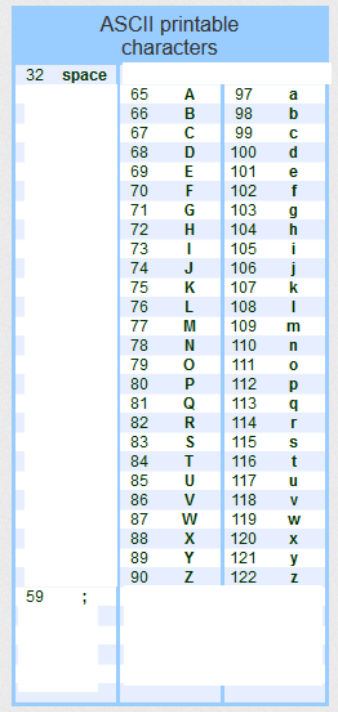


Figura 1: Caracteres possíveis para as informações que serão armazenadas na string.

A estrutura *tipoLogradouro* deve ser composta obrigatoriamente por:

- Um vetor de caracter chamado **tipo**, com *80* posições, que indica o tipo do logradouro (por exemplo: alameda, feira, setor, rua...), que possui caracteres que pertencem a tabela ASCII;
- Um vetor de caracter chamado **nome**, com *80* posições, que indica o nome do logradouro, que possui caracteres que pertencem a tabela ASCII;
- Um vetor de caracter chamado **complemento**, com *80* posições, que indica o complemento do logradouro, que possui caracteres que pertencem a tabela ASCII;

Restrições

A criação de outras funções auxiliares e o uso de outras bibliotecas são permitidos. Para este exercício, não é permitido o uso da biblioteca *string.h*

Ao enviar a sua solução pro MOJ, envie somente o arquivo com a extensão C com a função exigida do enunciado, as estruturas, a inclusão das bibliotecas utilizadas e as funções auxiliares (caso existam). Não inclua neste arquivo a função *main*.

Entrada

A sua função não fará nenhuma leitura da entrada padrão.

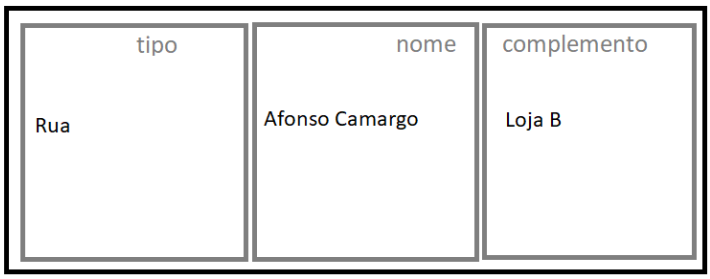
Saída

A sua função não fará nenhuma impressão na saída padrão.

Exemplos

Exemplo 1

Suponha que a sua estrutura esteja preenchida da seguinte forma:



A função deve preencher o segundo argumento da função da seguinte forma:

Rua;Afonso Camargo;Loja B

Exemplo 2

Suponha que a sua estrutura esteja preenchida da seguinte forma:

A função deve preencher o segundo argumento da função da seguinte forma:

Avenida;da Paz;

Problema: Adress

The file format *CSV*, **C**omma-**S**eparated-**V**alues, is a type of file that stores data separeted by comma. At Brazil, as much at other latin countries, decimals digits are also separed by *vírgula*, therefore, to not conflict, a *CSV* file created in a machine with Windows SO and located in Brazil uses the character *semicolon* to split the data. Each line of this file, when opened using programs such as Microsoft Excel or Google Sheet, will be transform in a line of the spreadsheet, and each column of the spreadsheet will have the data that is separeted by semicolon. For example, suposes that you are creating a table that the first column indicates the type of the adress, the second column the name of the adress and the third the adress complement.

Rua	Afonso Camargo	Loja B
Avenida	da Paz	

O *CSV* correspondente seria:

```
Rua;Afonso Camargo;Loja B
Avenida;da Paz;
```

Write a function called *criaLinhaCSV*, that receives a filled structure of type *tipoLogradouro*, that holds the information of the type of the adress, the name and the adress complement, and a string of length of 240, that must hold the information that the structure has, split by semicolon. For this problem, there is no need to validate the elements of the structure. Considers that the strings has only letters from A to Z, uppercase and lowercase, and spaces.

```
void criaLinhaCSV(struct tipoLogradouro info, char linha[240]);
```

Each argument of the function represents, respectively:

- The information of the adress, stored in a variavle of type *tipoLogradouro*, and;
- Where the information must be stored: the type, the name and the adress complement, respectively, seperated by semicolon , as the template <**TYPE**>;<**NAME**>;<**COMPLEMENT**>. Consider that this string is empty when is passed to the function.

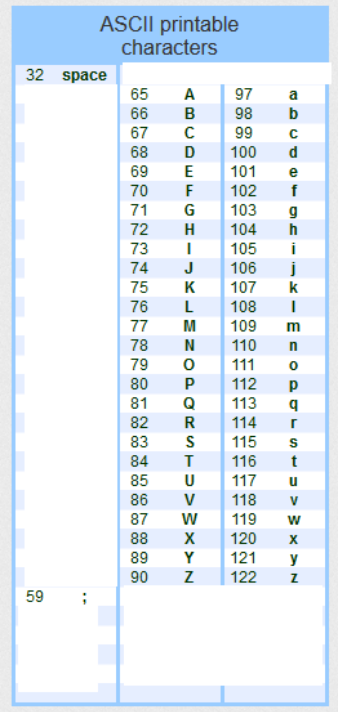


Figura 2: Possible characters.

The structure *tipoLogradouro* must have the folowing elements:

- A character's array called **tipo**, with size of *80*, that indicates the type of the adress (street, avenue, sector...), that has only characters in ASCII table;
- A character's array called **nome**, with size of *80*, that indicates the name of the adress, that has only characters in ASCII table and;
- A character's array called **complemento**, with size of *80*, that indicates the adress complement, that has only characters in ASCII table.

Restrictions

It is allowed to write other auxiliary functions. However, it is not allowed the use any of the functions from *string.h*.

To submit your solution to MOJ, send only the C file with the function that is request and the auxiliary functions (if they exist). Do not includes in this file the function *main*.

Input

There is no input for this problem.

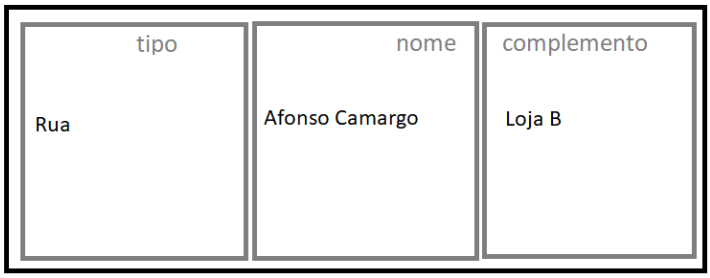
Output

There is no output for this problem.

Examples

Example 1

Suposes that your structure is filled as following:



Your function must store this information in the second argument of the function as following:

Rua;Afonso Camargo;Loja B

Exemplo 2

Suposes that your structure is filled as following:

Your function must store this information in the second argument of the function as following:

Avenida;da Paz;