

## LINGUAGEM DE PROGRAMAÇÃO

### RESUMO

Nesse resumo vou falar sobre a matéria de Linguagem de Programação; a Linguagem de Programação ou Linguagens de Programação são sintaxes usadas para transcrever comandos para a máquina. As máquinas utilizam bits de circuitos elétricos transferindo informações em binário para realizar uma ação, o que chamamos de linguagem de baixo nível ou Linguagem de Máquina. As Linguagens de Programação são sintaxes da linguagem humana, especialmente do inglês, para compilar comandos e traduzi-los para as máquinas, chamamos de linguagens de alto nível.

Escrevemos as linhas de código e comandos sob determinado paradigma, indentados e separados em arquivos. Com a linguagem C é estruturada para criar programas de computador, ela deve ser feita em uma IDE. Ao contrário de linguagens voltadas para Web onde o código-fonte pode ser escrito em um Editor de Texto e a visualização do programa se dá em um browser, C é construído em um IDE para criar programas de computador no desktop.

## PARADIGMAS DE PROGRAMAÇÃO

Paradigma de programação estrutura e torna disponível a projeção que um desenvolvedor poderá vislumbrar sobre a estruturação e execução de um software. Orientação à Objetos, por exemplo, faz seus recursos para criar abstrações de programa. Há dois tipos de Paradigmas de Linguagem:

- i) Linguagem Imperativa
  - (1) - Paradigma Procedural
  - (2) - Paradigma de Orientação a Objetos
- ii) Linguagem Declarativa
  - (1) - Paradigma Funcional
  - (2) - Paradigma Lógico
  - (3) - Paradigma Matemático
  - (4) - Paradigma Reativo

Java é uma linguagem que suporta apenas o paradigma de orientação a objetos. A linguagem C suporta o paradigma funcional e procedural, já outras linguagens são híbridas. Há também o paradigma estrutural, lógico e etc.

O Prolog é um exemplo de linguagem do paradigma lógico, muito usado na inteligência artificial e data-mining.

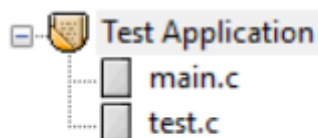
A primeira linguagem a ser estudada é a linguagem C, com o paradigma procedural ( procedure call ), procedimentos conhecidos pelas rotinas, sub-rotinas, métodos ou funções. O procedimento pode ser chamado a qualquer momento durante a execução do programa, inclusive por outros procedimentos.

## LINGUAGEM DE PROGRAMAÇÃO C

O exemplo de estudo será a Linguagem C, uma linguagem imperativa, de paradigma procedural e padronizada pela Organização Internacional de Padronização ( ISO ). A linguagem C se inicia com a chamada das bibliotecas, a mais padrão delas é chamado dos comandos de entrada e saída. A chamada das bibliotecas é feita pela hashtag e o include depois o nome da biblioteca entre as setas <>. Finalizando a linha com dois pontos.

```
#include <stdio.h>;
```

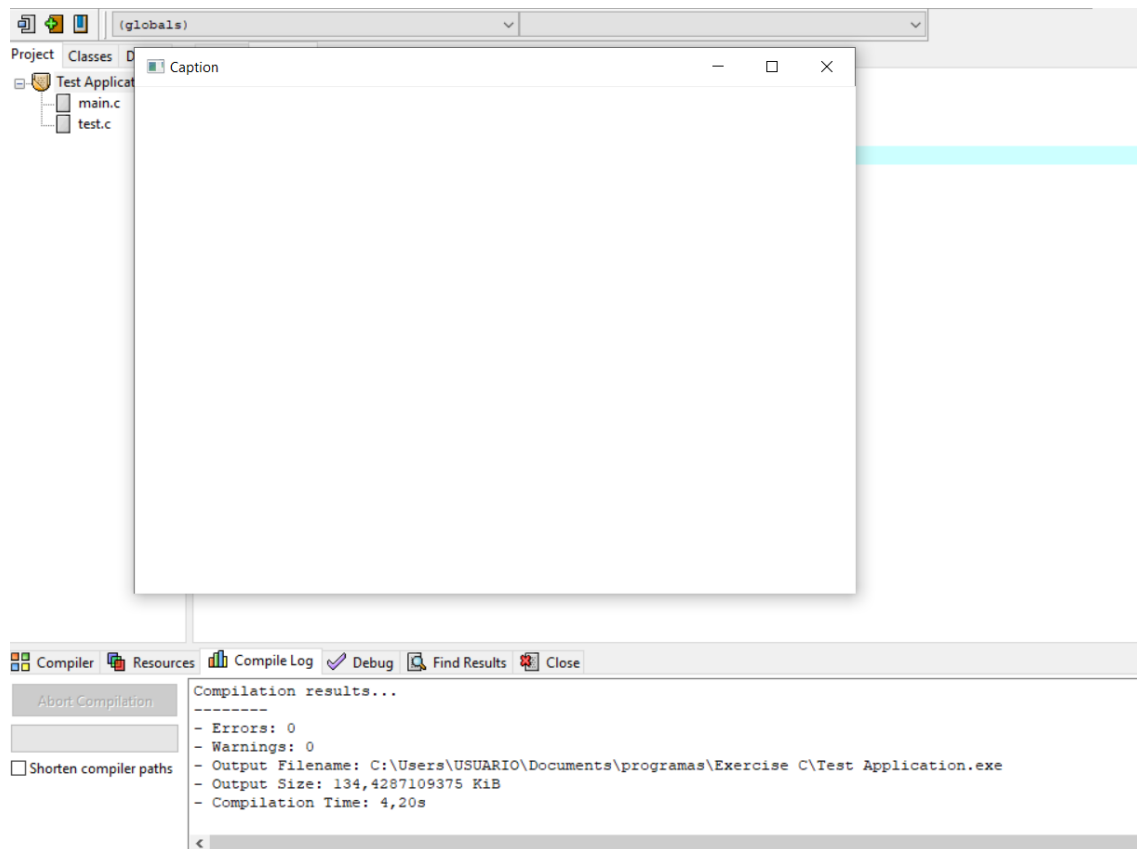
```
void main() {  
// Code here  
}
```



Depois vem a chamada na função principal, onde se transcreve os principais código-fonte do programa. Como toda linguagem C possui

uma vasta biblioteca de funcionalidades, nomenclatura de dados e os seus tipos, Comandos e operadores aritméticos. Os ponteiros, Registros – Struct, Union, Enum & Typedef, Ponteiros em Registros, Recursividade, Macros e a manipulação de arquivos e bits também fazem parte da estrutura em C. Vamos falar de cada uma delas separando-as. Começando com um resumo rápido dos comandos, dos dados e dos operadores.

A linguagem C tem muitas funcionalidades, mas geralmente é utilizada para construção de softwares desktops.



## Estrutura

A estrutura básica da linguagem em C começa por um cabeçalho como já foi dito, onde estará incluso as bibliotecas, declaração de variáveis, declaração de registros e funções e outros. Contando também com a nomenclatura de tipos de variáveis (tipos de dados):

*Int, Boolean, String, Float* e etc. Além dos Comandos básicos como: *If, For, While, Do* e outros. Uma estrutura deve ser declarada antes da função principal e depois das bibliotecas.

## PONTEIROS

Ponteiros são um registro de variável que armazena o endereço de outras variáveis na memória. Em situações onde é desejável fazer alocação dinâmica da memória, Manipular arrays e registros, referenciar pilhas, Listas, árvores e grafos. A sintaxe de um ponteiro é descrita como:

```
Int* ponteiro; = tipo do ponteiro* nome_do_ponteiro;
```

## REGISTROS: STRUCT, UNION, ENUM E TYPEDEF

Registros são estruturas que armazenam dados do mesmo tipo ou de tipos de diferentes.

Struct são registros que armazenam geralmente dados de tipos diferentes. É declarado um struct depois dando um nome para ele e depois exportando para outras partes do código. Uma struct se começa pela palavra reservada, em seguida o nome do dado e depois o nome da variável.

Para acessar uma struct, você deve colocar o nome\_da\_variável.nome\_do\_campo. O uso mais comum para struct são os vetores.

Union é uma estrutura de dados que permite armazenar dados de tipos diferentes no mesmo local da memória.

Enum são tipos definidos pelo usuário, utilizando uma lista de identificadores.

Typedef também podem ser utilizados para dar nomes complexos, como as estruturas.

Ponteiros também podem ser usados para no caso de registros. Como no caso já citado para alocação dinâmica da memória que é um gerenciamento manual da memória como recurso, usando comandos como malloc, realloc, calloc e free.

## RECURSIVIDADE E MACROS

Recursividade é quando a função a chama a própria a função. É um processo repetitivo de uma rotina. Iteração é uma chamada de repetição através de forma de laços, como o *For, Do, While...* Já a recursão é a repetição, mas em forma de chamadas para ela mesma. E Macros são atribuição de constantes na linguagem C, por essa sintaxe #define.

## ARQUIVOS E OPERAÇÕES COM BITS

Existem funções em C capazes de ler, abrir e executar arquivos. E também é possível fazer operações a nível de máquina em C.

Para leitura de arquivos: r

Para escrita de arquivos: w

Escrevendo ao final do arquivo: a

Abre um arquivo tanto para leitura quanto para escrita: +w

Quizzes - TESTE TÉCNICO – NÍVEL BÁSICO

Question 1

60 seconds

Q. O que significa o comando printf na linguagem C?

— answer choices —

- ☐ Pare
- ☐ Leia do teclado
- ☐ Imprima na tela
- ☐ Calcule

Printf faz parte de um comando de saída de dados. Imprime na tela ou no terminal o que é colocado.

Question 2

60 seconds

Q. O que significa o comando scanf na linguagem C?

— answer choices —

- ☐ Pare
- ☐ Leia do teclado
- ☐ Imprima na tela
- ☐ Calcule

O comando scanf é o comando de entrada de dados em C, que faz parte da biblioteca do stdio.h. scanf é “Leia do teclado”.

Question 3

60 seconds

Q. Qual a função do comando system("pause") na linguagem C?

— answer choices —

- ☐ Congele a tela de exibição
- ☐ Saia do programa.
- ☐ Finalize o programa.
- ☐ Altere a localidade do programa

A função system("pause") é uma função que é utilizada para congelar a tela de exibição.

Question 4

60 seconds

Q. O que significa #include<stdio.h> ?

— answer choices —

- ☐ Inserção de comandos
- ☐ Inclusão de biblioteca
- ☐ Declaração de Variáveis
- ☐ Pedir prioridade na execução

`#include <stdio.h>` o `#include` é uma inclusão de biblioteca para o arquivo e conseqüentemente para aplicação, enquanto `<stdio.h>` é a especificação da biblioteca, que no caso se tratada da biblioteca de comandos de entrada/ saída.

Question 5

60 seconds

Q. Para uma nota de um aluno qual tipo de dados devo utilizar?

— answer choices —

<input type="radio"/> char	<input type="radio"/> int
<input type="radio"/> float	<input type="radio"/> boolean

Notas escolares são, geralmente, números “quebrados” ou números reais. Usamos o tipo de dado float para atribuição desses números.

Question 6

60 seconds

Q. Para a idade de uma pessoa, qual tipo de dados devo utilizar?

— answer choices —

<input type="radio"/> char	<input type="radio"/> int
<input type="radio"/> float	<input type="radio"/> boolean

A idade das pessoas é dita por números inteiros, usamos o tipo int para esses números.

Question 7

60 seconds

Q. Em C, qual será o valor de r, de acordo com o trecho de código:

```
int a=2, b=2, c=4;  
r= a+ (b*c);
```


— answer choices —

<input type="radio"/> 16	<input type="radio"/> 12
<input type="radio"/> 10	<input type="radio"/> 8

A questão atribui três variáveis numéricas do tipo inteiro. a, b, c e um resultado r na qual se tem operação matemática que resultará no valor 10.

Question 8

120 seconds



Q. Assinale a alternativa CORRETA sobre a linguagem C:

answer choices

☐ A função `maim()` é uma função especial por onde o programa inicia sua execução. Contém os dados que comandam toda a execução do programa.

☐ Comentários em C devem iniciar com um `\*` e terminar com `*`, podendo começar em uma linha e terminar em outra. Ou usa-se `\\` no início da linha. Exemplo: `//Isto é um comentário`

☐ C retém a filosofia básica de que os programadores são iniciantes e não sabem o que estão fazendo.

☐ A linguagem C foi criada em 1972 por Dennis Ritchie, como uma linguagem para escrever sistemas operacionais. Objetivo: projetar o sistema Unix.

A questão seguinte é um pouco sobre a história da linguagem C criada em 1972 por Dennis Ritchie, como uma linguagem para escrever sistemas operacionais. Teve como objetivo projetar o sistema Unix.

Question 9

60 seconds

Q. A declaração de uma variável que seja capaz de armazenar o nome completo de um aluno é:

answer choices

☐ `char nome;`

☐ `int nome[50];`

☐ `string nome;`

☐ `char nome[50];`

A variável capaz de armazenar um nome completo é a variável da qual atribuímos o nome "nome" determinando o seu tamanho[50] e o seu tipo char.

Question 10

60 seconds

Q. Qual dos códigos abaixo corresponde uma saída/exibição CORRETA em linguagem C?

answer choices

☐ `print("Ola Mundo");`

☐ `printf("Ola Mundo");`

☐ `scanf("%s", "Ola Mundo");`

☐ `print("%s", "Ola Mundo");`



Por fim, a última questão é em relação ao código de saída e exibição na linguagem C. O comando `printf("Mensagem aqui");` é o comando correto para esse caso.

## Quizzes - TESTE TÉCNICO – NÍVEL INTERMEDIÁRIO

1. Size of a union is determined by size of the.

- ☐ First member in the union
- ☐ Last member in the union
- ☐ C. Biggest member in the union
- ☐ Sum of the sizes of all members

A UNION é uma estrutura de dados em C que permite agrupar dados do tipo diferentes como um único objeto de dados. São tipos de dados derivados que compartilham o mesmo espaço de armazenamento.

O tamanho de uma UNION é determinado pelo maior membro da UNION.

4. Which of the function cannot be a structure member?

- ☐ Another structure
- ☐ Function
- ☐ Array
- ☐ None

A função que não pode ser um membro de estrutura é uma função.

5. In a C Program if `int a = 5; int *ptr = &a;` then the `printf` statement with `&a, a, *ptr, ptr`

- ☐ address, Value, Value, Address
- ☐ address, value, value, value
- ☐ all address
- ☐ All values

Na questão acima, o enunciado entrega um ponteiro e pergunta o que significa cada palavra.

Os ponteiros são dados por `int*ptr = &a`; e na questão é perguntado o que é o `&a`, `a`, `*ptr` e o `ptr`. `&a` é atribuição do ponteiro, o endereço que o ponteiro ficará, o `a` é o valor, `*ptr` é o valor e `ptr` o nome do endereço.

### Questão 1/10 - Linguagem de Programação

Uma estrutura deve ser declarada em qual posição do programa:

Nota: 0.0

- ☐ A Dentro do programa main
- ☐ B Antes das bibliotecas
- ☒ C **Após as bibliotecas e antes do main**
- ☐ D Depois das bibliotecas

A questão 1/10 Pergunta-se onde deve ser declarada uma estrutura(struct); A resposta correta é a letra C.

```
main.c
2  #include <stdlib.h>
3  #include <locale.h>
4  #include <conio.h>
5  // Global Scopo
6  struct ficha_de_aluno
7  {
8      char nome[50];
9      char disciplina[30];
10     float nota_prova1;
11     float nota_prova2;
12 };
13 struct ficha_de_aluno aluno;
14
15 void main() {
16     setlocale(LC_ALL, "Portuguese");
17     //Local Scopo
18     printf("\n----- Cadastro de aluno ----- \n\n\n");
19
20     printf("Nome do aluno .....: \n");
21     fflush(stdin);
22     fgets(aluno.nome, 40, stdin);
```

#### Questão 2/10 - Linguagem de Programação

Na manipulação de arquivos, podemos abrir um arquivo de diversos modos de acesso. O modo de acesso é uma string que contém uma sequência de caracteres que informam se o arquivo será aberto para escrita ou leitura. Depois que abrir o arquivo, podemos executar os tipos de ação previstos pelo modo de acesso. Assim, não será possível ler de um arquivo que foi aberto somente para escrita.

Acerca dos modos de acesso usados na linguagem C, assinale a alternativa que contém todos eles.

- I. 'r' - Leitura de arquivo
- II. 'w' - Escrita em arquivo
- III. 'a' - Escrevendo ao final do arquivo (anexando)
- IV. 'w+' - Abre um arquivo tanto para leitura quanto para escrita. Se o arquivo já existir terá seu conteúdo substituído.
- V. Todas as alternativas estão corretas.

Na manipulação de arquivos, como já vimos no resumo: leitura de arquivos "r", escrita "w", escrita ao final do arquivo "a", Para abrir um arquivo tanto para leitura quanto para a escrita "w+".

#### Questão 3/10 - Linguagem de Programação

O uso de funções recursivas é uma prática bastante comum em programação. Acerca de recursividade e funções recursivas, assinale a alternativa correta.

Recursividade da função é quando uma função chama ela mesma, porém quando não há controle pode causar um loop infinito, acarretando assim um travamento no sistema por inteiro.

#### Questão 4/10 - Linguagem de Programação

Considerando a definição da struct livro, a seguir. Indique nas alternativas qual define uma variável deste tipo corretamente.

```
struct Livros{  
    char titulo[50];  
    char autor[50];  
    char assunto[100];  
    int  codigo;  
};
```

Para definir uma struct corretamente devemos marcá-la struct depois o nome da struct e depois a variável.

```
6 struct ficha_de_aluno  
7 {  
8     char nome[50];  
9     char disciplina[30];  
10    float nota_prova1;  
11    float nota_prova2;  
12 };  
13 struct ficha de aluno aluno;
```

Nesse caso dos livros seria struct Livros príncipe;

---

#### Questão 5/10 - Linguagem de Programação

As funções *memset()*, *memcpy()*, *memmove()* e *memcmp()* pertencem a qual biblioteca na linguagem C?

As funções:

1. Memset()
2. Memcpy()
3. Memmove()
4. Memcmp()

Fazem parte da biblioteca *string.h* da linguagem C para manipular strings, tendo ações como: copiar, concatenar, descobrir e comparar.

---

#### Questão 6/10 - Linguagem de Programação

Marque a alternativa correta referente ao acesso de um campo da *struct*.

Para acessar um campo da string devemos colocar o nome da variável da struct e depois o nome do campo.

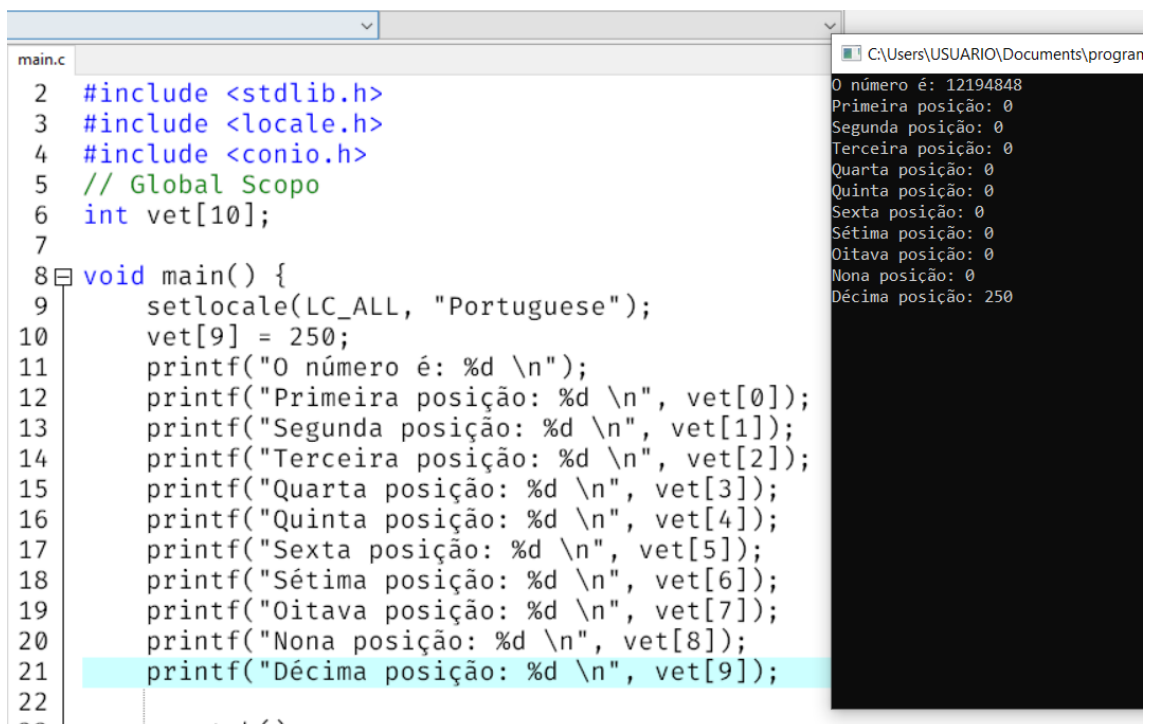
```
printf("Nome: %s", aluno.nome);
```

#### Questão 1/12 - Linguagem de Programação

Um vetor pode ser definido como um conjunto de elementos consecutivos, do mesmo tipo, que podem ser acessados, individualmente a partir de um único nome.

Com relação a definição de vetores na linguagem C, assinale a alternativa que corresponde a forma correta de atribuir o valor 250 na última posição de um vetor chamado vet do tipo inteiro de 10 posições:

Partindo do princípio que a contagem em computação começa do zero. A décima posição é a última posição, portando vet[9].



The screenshot shows a C program in a code editor. The program defines an integer array 'vet' of size 10 and prints the value at each index from 0 to 9. The value at index 9 is set to 250. The output window shows the following results:

```
0 número é: 12194848
Primeira posição: 0
Segunda posição: 0
Terceira posição: 0
Quarta posição: 0
Quinta posição: 0
Sexta posição: 0
Sétima posição: 0
Oitava posição: 0
Nona posição: 0
Décima posição: 250
```

Com relação as funções de Strings, analise atentamente o código em C apresentado abaixo:

```
#include <stdio.h>
#include <string.h>

int main() {
    char str1[100], str2[100], str3[100];
    printf( "Entre com uma string: " );
    gets( str1 );
    strcpy( str2, str1 );
    strcpy( str3, "Veja o que foi armazenado: " );
    printf( "\n\n%s%s", str3, str2 );
    return(0);
}
```

Com base no código apresentado acima, assinale a alternativa que apresenta a saída correta deste programa:

A questão acima é a manipulação de strings daquela biblioteca da string.h. Na primeira linha depois da abertura da função principal é armazenamento dos dados do tipo char, chamando-os de str1, str2, str3.

Logo depois ele irá gravar o valor em str1 que o usuário digitar e fazer uma cópia em str2 com o comando strcpy. Nas linhas seguintes ele grava o str3 como : “Veja o que foi armazenado”, dando por final o printf mostrando o str3 e str2. O resultado final é “Veja o que foi armazenado: valor”.

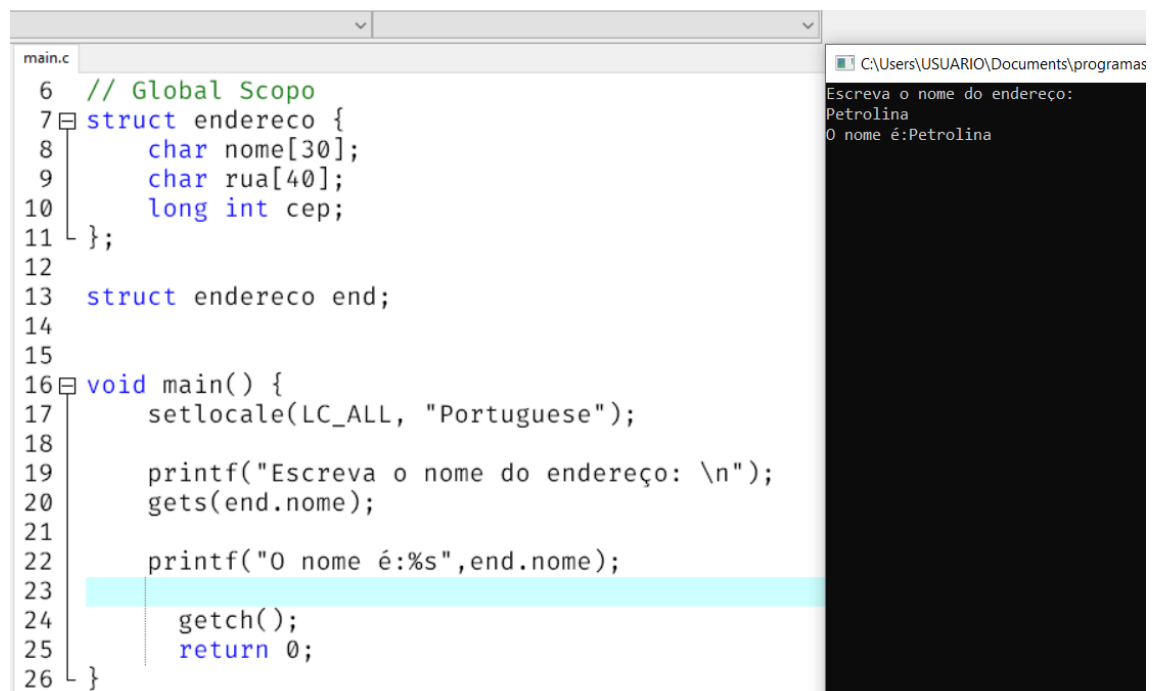
Neste contexto, analise atentamente a instrução apresentada abaixo:

```
struct endereco{
    char nome[30];
    char rua[40];
    long int cep;
};
```

```
gets(end.nome);
```

Assinale a alternativa que corresponde ao resultado obtido após a execução do fragmento de código contendo a instrução gets apresentada acima,

A função gets grava dados que o usuário digitar, nesse caso irá guardar na struct endereço no campo nome.



The screenshot shows a C program in a code editor and its execution output. The code defines a struct named 'endereco' with fields 'nome' (30 chars), 'rua' (40 chars), and 'cep' (long int). In the main function, it sets the locale to Portuguese, prompts the user to enter an address name, reads the input using 'gets', and prints the stored name. The output window shows the prompt 'Escreva o nome do endereço:', the user input 'Petrolina', and the program output 'O nome é: Petrolina'.

```
main.c
6 // Global Scope
7 struct endereco {
8     char nome[30];
9     char rua[40];
10    long int cep;
11 };
12
13 struct endereco end;
14
15
16 void main() {
17     setlocale(LC_ALL, "Portuguese");
18
19     printf("Escreva o nome do endereço: \n");
20     gets(end.nome);
21
22     printf("O nome é:%s", end.nome);
23
24     getch();
25     return 0;
26 }
```

C:\Users\USUARIO\Documents\programas

Escreva o nome do endereço:  
Petrolina  
O nome é: Petrolina