

## Problema A

Australopithecus está aprendendo a sua letra favorita ("r") em seus travas-línguas preferidos! Ele então pediu sua ajuda para fazer um programa que lê um trava-língua e conte quantas vezes a letra "r" aparece nela.

### Entrada

A entrada é composta por uma única linha contendo um texto com, no máximo, 100 caracteres.

### Saída

Um inteiro que representa o número de vezes que o caractere "r" aparece no texto.

Exemplo de entrada	Exemplo de saída
Trazei tres pratos de trigo para tres tigres tristes comerem	9

## Problema B

Joãozinho é um jovem entusiasta de enigmas e decidiu desafiar seus amigos com um jogo curioso: encontrar palavras ou frases que são palíndromos. Ele explicou que um palíndromo é uma sequência de caracteres que pode ser lida da mesma forma da direita para a esquerda, como da esquerda para a direita, sempre com o mesmo sentido. Por exemplo, a palavra “mirim” é um palíndromo, mas “lua” não é.

Para tornar o jogo ainda mais interessante, João quer criar um programa que automatize essa verificação. Ele escreveu uma função main, e precisa com que você faça o trecho de código da verificação:

A main feita por Joãozinho:

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

int main(){
    char string[32];
    fgets(string, 32, stdin);
    int tamanho = strlen(string);
    int eh_palindromo;

    //TRECHO DE CODIGO AQUI

    if(eh_palindromo){
        printf("A string eh um palindromo.\n");
    } else {
        printf("A string nao eh um palindromo.\n");
    }
    return 0;
}
```

### Entrada

A primeira e única linha da entrada é composta por uma única string de, no máximo, 30 caracteres, que pode conter letras minúsculas e espaços.

### Saída

Seu programa deverá imprimir se a string passada é um palíndromo ou não.

Exemplo de entrada	Exemplo de saída
mirim	A string eh um palindromo.

Exemplo de entrada	Exemplo de saída
brasil	A string nao eh um palindromo.

Exemplo de entrada	Exemplo de saída
abc cba	A string eh um palindromo.

Exemplo de entrada	Exemplo de saída
a ba	A string nao eh um palindromo.

**OBS:** Lembre-se que a função fgets() adiciona um caractere '\n' ao final da string lida da entrada.

## Problema C

Joana adora brincar com strings e criar padrões curiosos. Em um de seus desafios, ele decidiu pegar duas strings e concatenar o inverso da primeira string ao final da segunda. Agora, ela quer automatizar esse processo com a sua ajuda!

Implemente um trecho de código que concatena o inverso da primeira string ao final da segunda:

Para testar, você deve completar a seguinte main:

```
int main(){
    char string1[17], string2[34];
    scanf("_____", _____, _____);
    int tam1 = _____;
    int tam2 = _____;

    //TRECHO DE CODIGO AQUI

    printf("%s\n", string2);
    return 0;
}
```

### Entrada

A entrada é composta por uma única linha, contendo duas strings, separadas por espaço.

### Saída

A segunda string concatenada com o inverso da primeira.

Exemplo de entrada	Exemplo de saída
carro bola	bolaorrac

## Problema D

Uma empresa deseja otimizar a forma de ler o feedback de seus clientes, após testes passados, notou-se que a informação mais importante tende a ficar **na segunda frase**, ou seja, entre o ponto final da primeira frase e o ponto final da segunda frase. A empresa já possui a main, você deve implementar o trecho de código que encontra a substring desejada:

A seguir o código da main incompleto:

```
int main(){
    char s[300];
    char sub[300];
    fgets(s, 300, stdin);

    //TRECHO DE CODIGO AQUI

    printf("%s\n", sub);
    return 0;
}
```

### Entrada

A entrada é composta por um texto que possui ao menos duas frases e que termina com “\n”. A entrada nunca possui mais de 300 caracteres e sua substring também não passa de 200.

### Saída

Seu programa deverá imprimir a substring entre o primeiro ponto final e o segundo ponto final. Lembrando que o espaço após o primeiro ponto final deve ser ignorado.

Exemplo de entrada	Exemplo de saída
Recebi o produto antes do prazo. O produto apresenta qualidade razoável.	O produto apresenta qualidade razoável.