

Trabalho 01 - Concatenação de Strings

Data de entrega: 22/09/2021

Importante:

- **Não** olhe códigos de outros ou da internet. Exceto o que é fornecido. Também não mostre ou publique o seu.
- Em caso de plágio, fraude ou tentativa de burlar o sistema será aplicado nota 0 na disciplina aos envolvidos.
- Alguns alunos podem ser solicitados para explicar com detalhes a implementação.
- Passar em todos os testes do run.codes não é garantia de tirar a nota máxima. Sua nota ainda depende do cumprimento das especificações do trabalho, qualidade do código, clareza dos comentários, boas práticas de programação e entendimento da matéria demonstrada em possível reunião.
- Você deverá submeter, até a data de entrega, o seu código na plataforma run.codes.

Este trabalho é individual e deverá ser implementado em linguagem C. Você deverá implementar uma função que recebe três “strings” e concatena as duas primeiras na terceira. Geralmente, em C, uma “string” é um arranjo/vetor de **char**, que tem um tamanho máximo fixado, porém o número de caracteres utilizados em um determinado momento pode variar e o final da string é indicado por um caractere especial ‘\0’. Considere a seguinte declaração de uma “string”:

```
char frase1[20];
```

Essa declaração reserva na memória um espaço para 20 caracteres, porém é recomendado guardar pelo menos um caractere para o final de string ‘\0’. Poderíamos então alocar a frase “banana boa” nessa string e na memória teríamos algo assim:

char * frase1



Lembre que um arranjo de chars (**char** frase1[20]) é também um ponteiro para o primeiro char daquele arranjo, ou seja frase1 pode ser compreendido como um **char*** que tem o seu valor (ou seja, o endereço da memória que reservamos) fixo.

A sua função deve concatenar as duas frases e colocar um espaço entre elas, além disso sua função devolverá duas informações: o número total de caracteres na string resultante e também o número de espaços (entre as palavras) no resultado. Como as funções em C só permitem o retorno de um valor, o número de espaços deverá ser passado por uma variável

que sua função receberá como referência. As strings também serão recebidas por referência. Dessa forma a sua função terá a seguinte assinatura:

int concatena(**char*** s1, **char*** s2, **char*** sres, **int*** espacos);

Note que tanto s1, como s2 e sres já devem ser alocadas fora da função e apenas o endereço de memória das três “strings”, assim como da variável para armazenar o número de espaços.

Você receberá uma implementação parcial do programa, que já faz a leitura das strings. Nessa implementação é usado a função fgets que tem a seguinte assinatura:

char* fgets (**char*** str, **int** num, **FILE*** stream);

A função fgets lê uma string, de até num - 1 caracteres, e a copia em str, essa leitura é feita em *stream*, que pode ser tanto um arquivo quanto a entrada padrão do sistema “stdin”. A string é lida até ser encontrado uma quebra de linha ‘\n’ ou um final de arquivo. A quebra de linha é um caractere válido, e portanto ele é colocado em str.

Para exemplificar considere a seguinte declaração:.

```
char frase1[20];
char frase2[20];
char fraseResultado[40];
fgets(frase1, 20, stdin);
fgets(frase2, 20, stdin);
```

E considere o seguinte exemplo de entrada:

```
Algoritmos e
Estr. de Dados.
```

O resultado seria:

frase1:

A	l	g	o	r	i	t	m	o	s		e	\n	\0						
---	---	---	---	---	---	---	---	---	---	--	---	----	----	--	--	--	--	--	--

frase2:

E	s	t	r	.		d	e		D	a	d	o	s	.	\n	\0			
---	---	---	---	---	--	---	---	--	---	---	---	---	---	---	----	----	--	--	--

A sua concatenação deverá ignorar as quebras de linhas e deverá colocar um único caracter de fim de string ‘\0’ ao final do resultado. A implementação parcial também já imprime o resultado da concatenação e a informação do número de caracteres e espaços, no exemplo acim será impresso:

Algoritmos e Estr. de Dados.
String com 28 caracteres e 4 espaços.

Resumindo:

- Você não deve usar nenhuma biblioteca especializada.
- Você não deve mexer na função main, apenas na função “concatena”, mas pode criar outras funções se precisar.
- Você deverá implementar a função que recebe 3 strings (apontadores para arranjos de chars), e um apontador para um inteiro.
- Você deverá concatenar a primeira com a segunda string, além de um espaço entre elas, e colocar o resultado na terceira string.
- Você deverá colocar na variável que você recebe o apontador o número de espaços encontrados.
- Sua função deve retornar o número de caracteres (sem contar o caracter ‘\0’)