

Linguagem C:

Concatenando um caractere a uma string

Como concatenar um caractere a uma string? Tente fazer isto em diversas linguagens de programação e depois tente fazer em Linguagem C. Você verá que esta é uma tarefa das mais difíceis de serem realizadas. Se não acredita, tente!

Para facilitar nossa vida, achei um código escrito por um italiano que funciona muito bem e, como demorei bons minutos para resolver o problema, resolvi escrever este pequeno artigo para nunca mais esquecer da solução.

Vejamos o código-fonte:

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

#define ERR_MESSAGE__NO_MEM "Not enough memory!"
#define allocator(element, type) _allocator(element, sizeof(type))

/** Allocator function (safe alloc) */
void *_allocator(size_t element, size_t typeSize)
{
    void *ptr = NULL;
    /* check alloc */
    if((ptr = calloc(element, typeSize)) == NULL)
    {printf(ERR_MESSAGE__NO_MEM); exit(1);}
    /* return pointer */
    return ptr;
}

/** Append function (safe mode) */
char *append(const char *input, const char c)
{
    char *newString, *ptr;

    /* alloc */
    newString = allocator((strlen(input) + 2), char);
    /* Copy old string in new (with pointer) */
    ptr = newString;
    for(; *input; input++) {*ptr = *input; ptr++;}
    /* Copy char at end */
    *ptr = c;
    /* return new string (for dealloc use free().) */
    return newString;
}

/** Program main */
int main (int argc, const char *argv[])
```

Dicas de Programação

Prof. Edwar Saliba Júnior

```
{
    char *input = "Ciao Mondo"; // i am italian :), this is "Hello World"
    char c = '!';
    char *newString;

    newString = append(input, c);
    printf("%s\n", newString);
    /* dealloc */
    free(newString);
    newString = NULL;

    exit(0);
}
```

```
newString is [C] [i] [a] [o] [\32] [M] [o] [n] [d] [o] [!] [\0]
```

Fonte: Stackoverflow <https://stackoverflow.com/questions/10279718/append-char-to-string-in-c?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa>.

Acesso em: 20 maio 2018.

E para completar sua resposta, o italiano escreveu:

“Não altere o tamanho da matriz (`[len + 1]`, etc.) sem saber seu tamanho exato, isso pode danificar outros dados. Aloque uma matriz com o novo tamanho e coloque os dados antigos dentro, lembre-se de que, para uma matriz `char`, o último valor deve ser `\0`; `calloc()` define todos os valores como `\0`, o que é excelente para matrizes de caracteres. Eu espero que isso ajude.”¹

Este texto e código-fonte foram retirados do site de dúvidas e respostas, também conhecido como fórum, chamado Stackoverflow. Um dos melhores da área de programação. Esta resposta dada pelo italiano funcionou perfeitamente para o problema que eu tinha em mãos e, para os superticiosos, uma incrível coincidência que percebi: o texto foi escrito no dia 20 de maio de 2014 e eu o achei e usei no dia 20 de maio de 2018.

¹ Para a tradução do texto do Inglês para o Português foi utilizada a ferramenta Google Translator <<https://www.google.com/search?q=google+tradutor&ie=utf-8&oe=utf-8&client=firefox-b>> em 21 maio 2018.