

Mais informações em <https://edisciplinas.usp.br/mod/forum/discuss.php?d=898002https://edisciplinas.usp.br/mod/forum/discuss.php?d=898002>

char * ep1Passo2XOR(char *, int *)

- Essa função vai receber uma string correta e um vetor de inteiros e devolver uma nova string resultante da tarefa de calcular e concatenar os XOR. Aqui podemos ter um problema pois dentre os novos valores a serem concatenados pode ser que tenha um zero e dessa forma não seria possível, a partir desse ponto, sabermos o tamanho da string corretamente com a função strlen. Por isso essa função precisa de uma mudança. Vamos ter que incluir um novo argumento que vai devolver o tamanho da string resultante. Dado que essa string resultante não respeita mais a propriedade de ter um 0 marcando o final, seria ok a gente ter uma string com de fato o tamanho útil da string, considerando apenas os caracteres e sem contar esse último zero. Mas isso pode causar confusão porque a gente teria que saber quando a string possui um item a mais e quando não possui. Para evitar essa confusão, vamos considerar que as strings sempre vão ter um caractere a mais, mesmo quando esse caractere não esteja de fato sendo utilizado (mas sempre coloquem ele - o zero - no final). Então, essa função precisa passar a ser assim:

char * ep1Passo2XOR(char *, int *, int *): recebe uma string que tenha sido gerada como saída do Passo 1 do EP1, junto com um vetor Mágico de 256 inteiros de 0 a 255 sem repetição e um ponteiro para um inteiro que armazenará o tamanho da nova string a ser devolvida (Assim como strlen, esse tamanho não conta um 0 no final, mas esse zero estará lá e a área de memória precisará ter espaço para guardá-lo). A função devolve uma nova string correspondente à execução do Passo 2 do EP1. Se os argumentos não corresponderem ao que se espera, retorna NULL

char * ep1Passo3Comprime(char *)

- Essa função vai receber a string do passo anterior e devolver uma nova string de 48 caracteres (No caso, a string resultante precisará ter 49 posições seguindo o que comentei na função anterior para respeitarmos as regras de strings mesmo que agora a gente possa ter um 0 no meio do caminho). Para que a função funcione corretamente ela precisa saber o tamanho da string que ela está recebendo e para isso ela precisa ter um novo argumento que vai ser esse tamanho. Como a saída tem um tamanho fixo (49), a gente não precisa devolver o tamanho da nova string como fizemos na função anterior. Então essa função precisa passar a ser assim:

char * ep1Passo3Comprime(char *, int): recebe uma string que tenha sido gerada como saída do Passo 2 do EP1 e o tamanho dessa string, sem contar com o zero final dela, e devolve uma nova string de exatamente 48 caracteres correspondente à execução do Passo 3 do EP1. Se os argumentos não corresponderem ao que se espera, retorna NULL;