OBJETIVO: Exercitar a manipulação de Strings em C através da implementação de um sistema de geração de palavras (dicionário) a partir de um alfabeto.

QUESTÃO ÚNICA

Dado um tamanho máximo das palavras e um conjunto de caracteres (alfabeto), seu programa deverá gerar todas as palavras possíveis usando o conjunto de caracteres até o tamanho máximo definido.

Curiosidade:

Esta enumeração explícita de todas as palavras possíveis de um alfabeto é muito usada para descoberta de senhas. Tal ataque é conhecido como ataque de força-bruta, em que todas as senhas possíveis são testadas em um sistema de autenticação. Extremamente eficiente até um certo tamanho de senha (e.g., senhas até 6 caracteres), mas inviável para senhas maiores, devido à grande quantidade de possibilidades (exponencial).

O tamanho máximo das palavas será passado no primeiro argumento da execução do seu programa (`argv[1]`).

O conjunto de caracteres (alfabeto) será passado no segundo argumento da execução (`argv[2]`) e será representado por uma string, em que cada letra da string é um caractere que faz parte do conjunto de caracteres. Por exemplo, a string `"abc"` gerará o conjunto de caracteres formado pelas letras `a`, `b` e `c`. Neste trabalho, há a diferenciação de maiúsculas e minúsculas, ou seja, a string `"abcABC"` gerará o conjunto de caracteres formados pelas letras `a`, `b`, `c`, `A`, `B`, `C`. Números e caracteres especiais também poderão fazer parte da string.

Exemplo de Execução:

```
$ ./dicionario 3 'abc'
a
b
c
aa
ab
ac
ba
bb
bc
ca
cb
cc
aaa
aab
aac
aba
abb
abc
aca
acb
acc
baa
bab
bac
bba
bbb
bbc
bca
bcb
bcc
caa
cab
cac
cba
cbb
cbc
cca
ccb
ccc
```

DICAS

Na seguinte referência:

> Flor, Nick V. and Shannon, Haile (2011) "Technology Corner: Brute Force Password Generation -- Basic Iterative and Recursive Algorithms," Journal of Digital Forensics, Security and Law: Vol. 6 : No. 3 , Article 7.
>
> PDF: https://commons.erau.edu/cgi/viewcontent.cgi?article=1102&context=jdfsl

é explicado como implementar esse algoritmo de duas formas diferentes: usando iteração e usando recursão. Você pode se basear nele, ou implementar sua própria solução. Sua saída não precisa estar na mesma ordem que o exemplo de execução anterior. Mas todas as palavras precisam ser listadas.

Na referência acima, a implementação usando *recursão* é muito mais fácil e prática de se implementar. Note também que as palavras geradas pela função possuem exatamente o tamanho pedido no parâmetro dela. Como neste trabalho, está sendo pedido todas as palavras “até” o tamanho pedido, você precisará executar esta função (no seu *main*) para cada um dos tamanhos.

ENTREGA DO LABORATÓRIO

Envie, até 08/12/2022 às 23:59, o código-fonte para horacio@icomp.ufam.edu.br com o assunto “Entrega do 5o Laboratório de LPA”.