

## Exercício de Programação 1 – TipoString (Revisão de Programação e Tipos Abstratos de dados)

Valor: 5 pontos

Data de devolução: 17/08/2017

O objetivo desse trabalho é rever conceitos básicos de programação bem como explorar os conceitos de Tipos Abstratos de Dados (TADs).

Basicamente, você deverá criar um tipo abstrato de dados chamado **TipoString** e utilizá-lo em um programa que faz a criptografia de um texto usando a *Cifra de Vigenère*.

O TAD TipoString foi desenvolvido para representar strings compostas apenas de caracteres do conjunto [A..Z]. Considere que a sua string pode ter um tamanho máximo de 50 caracteres (não é necessário utilizar alocação dinâmica). Você deverá implementar as seguintes operações em seu TAD:

1. `TipoString CriaString(char *str);`  
Cria e retorna uma string contendo o texto passado como parâmetro. Caso algum caractere de `str` não seja válido, retorna NULL.
2. `TipoString CriaStringVazia();`  
Cria uma string vazia
3. `void ImprimeString(TipoString str);`  
Imprime uma string na saída padrão (tela)
4. `char GetChar(TipoString str, int i);`  
Retorna o caractere na posição `i` da string (considere que a string começa na posição 0). Caso o índice seja inválido, retorna o caractere '!'.  
`!`
5. `int SetChar(TipoString *str int i, char c);`  
Altera o caractere da posição `i` da string pelo caractere `c` passado como parâmetro. Retorna 0 em caso de sucesso ou 1 caso o índice ou o caractere sejam inválidos.
6. `int InsereChar(TipoString *str, char c);`  
Insere o caractere `c` no final da string aumentando o seu tamanho. Retorna 0 em caso de sucesso ou 1 caso a string já esteja com o seu tamanho máximo.
7. `int GetTamanho(TipoString str);`  
retorna o tamanho da string;

Como explicitado em cada caso, as operações implementadas deverão fazer testes de consistência, por exemplo se os caracteres fornecidos no `CriaString`, `SetChar` e `InsereChar` estão na faixa [A..Z], se os índices são válidos, etc.

Após o desenvolvimento do TAD, você deverá fazer um programa simples para testar as funcionalidades do seu TAD. Durante a correção, iremos testar o seu TAD com outros programas. Portanto, não faça nenhuma alteração na assinatura (cabeçalho) das funções descritas na página anterior.

Além disso você deverá fazer um programa que faz a criptografia de mensagens usando a *Cifra de Vigenère*. Esse método de criptografia consiste em substituir cada caractere  $c$  da string de entrada pela letra  $c + K$ , onde  $K$  é um valor fornecido por uma chave de criptografia e varia à medida que processamos a mensagem. A cifra de criptografia também é uma string com as letras representando números ( $A=1, B=2, \dots, Z=26$ ). Por exemplo, para a string `AEDSII` e a cifra `ABCD` teríamos a seguinte mensagem cifrada: `BGGWJK`. A forma mais fácil de visualizar isso é alinhar o texto e a chave repetindo-a quantas vezes for necessário e “somar” os dois textos:

```
AEDSII
ABCDAB
-----
BGGWJK
```

Caso alguma letra da nova mensagem fique fora do alfabeto (A..Z), volta-se ao início do alfabeto (aritmética modular):  $Z+1 = A$ ;  $Z+2=B$ , etc. Por exemplo, para a string `ABCXYZ` e cifra `ABZ`:

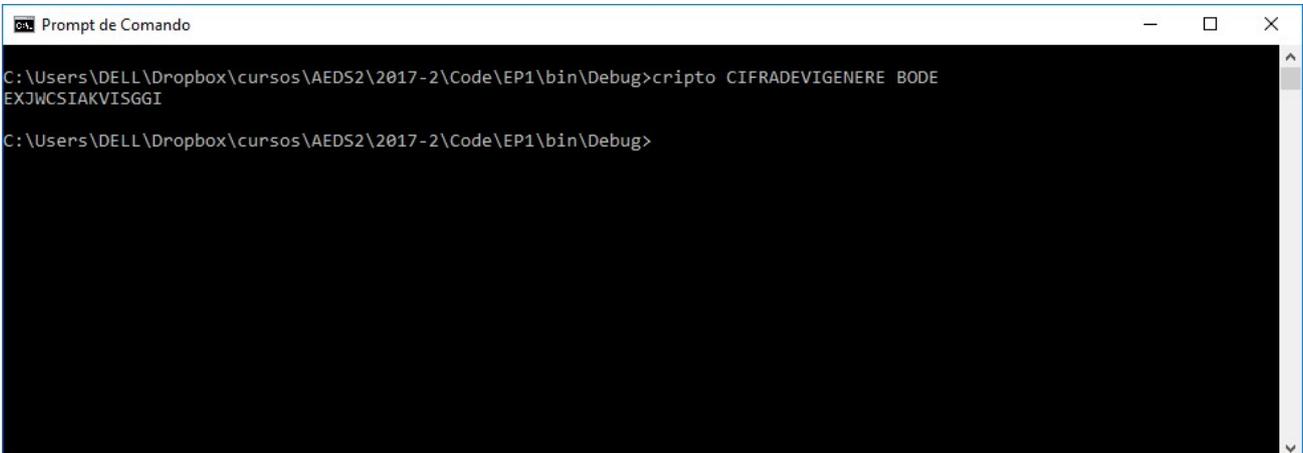
```
ABCXYZ
ABZABZ
-----
BDCYAZ
```

Outro exemplo segue abaixo para o texto `CIFRADEVIGENERE` e chave `BODE`:

```
CIFRADEVIGENERE
BODEBODEBODEBOD
-----
EXJWCSTIAKVISGGI
```

Tanto o texto a ser criptografado quanto a chave devem ser passados como parâmetros de linha de comando para o seu programa (veja a documentação sobre `argc` e `argv`). Dessa forma, se o seu programa se chama `cripto.exe`, o texto é `CIFRADEVIGENERE` e a chave `BODE`, a chamada vai ser feita da seguinte forma em um prompt do sistema:

```
%> cripto CIFRADEVIGENERE BODE
```



```

C:\Users\DELL\Dropbox\cursos\AEDS2\2017-2\Code\EP1\bin\Debug>cripto CIFRADEVIGENERE BODE
EXJWCSTIAKVISGGI
C:\Users\DELL\Dropbox\cursos\AEDS2\2017-2\Code\EP1\bin\Debug>
```

**O que deve ser entregue:**

- 1 Código fonte (.c e .h) do seu TAD. Para facilitar a correção eles devem ser nomeados `TipoString.c` e `TipoString.h`
- 2 Código fonte do programa feito para testar o seu TAD. Ele deve ser nomeado `EP1.c`
- 3 Código fonte do programa de criptografia. Ele deve ser nomeado `cripto.c`

Instruções específicas sobre a submissão serão fornecidas no moodle.

**Comentários Gerais:**

- 1 Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
- 2 Clareza, indentação e comentários no programa são fundamentais e serão avaliados.
- 3 Não utilize bibliotecas específicas de algum sistema operacional e que não façam parte da distribuição padrão da linguagem C.
- 4 O trabalho é individual
- 5 Trabalhos copiados serão penalizados conforme anunciado na primeira aula.
- 6 Penalização por atraso: 1 dia 50% da nota. 2 dias: 100% da nota.