

Alunos: Eduardo Cabral, Romulo Menezes e Victor Lopes

## Relatório 1

### Introdução

Optamos por desenvolver uma aplicação que descriptografa uma mensagem. Dentre os diversos tipos de criptografia, escolhemos a Cifra de César que é uma das mais antigas e mais simples forma de criptografar uma mensagem. Essa criptografia consiste em modificar uma letra do alfabeto por outra.

Para a nossa aplicação, ela decodifica um arquivo do tipo txt chamado 'arquivo.txt' que as letras estejam deslocadas três posições para frente, ou seja, 'a' vira 'd', 'b' vira 'e' e assim por diante e quando chega no 'x' ele vira 'a', 'y' vira 'b' e 'z' vira 'c'.

### Código

Na parte de declarações do programa lex, foram definidos tokens de cada letra do alfabeto com o seu determinado valor. Veja no código a seguir:

```
%option noyywrap
```

```
TK_A    d
```

```
TK_B    e
```

```
TK_C    f
```

```
TK_D    g
```

```
TK_E    h
```

```
TK_F    i
```

```
TK_G    j
```

```
TK_H    k
```

```
TK_I    l
```

```
TK_J    m
```

```
TK_K    n
```

```
TK_L    o
```

```
TK_M    p
```

```
TK_N    q
```

```
TK_O    r
```

```
TK_P    s
```

```
TK_Q    t
```

```
TK_R    u
```

```
TK_S    v
```

```
TK_T    w
```

```
TK_U    x
```

```
TK_V    y
```

```
TK_W    z
```

```
TK_X    a
```

```
TK_Y    b
```

```
TK_Z    c
```

Já na parte de regras de tradução do programa, foram definidas regras para caso um dessas letras que represente um token seja encontrada, o programa imprime no terminal a letra decodificada e assim forma a palavra a ser descriptografada. Veja o código a seguir:

```
[" "]    {printf(" ");}
{TK_A}   {printf("a");}
{TK_B}   {printf("b");}
{TK_C}   {printf("c");}
{TK_D}   {printf("d");}
{TK_E}   {printf("e");}
{TK_F}   {printf("f");}
{TK_G}   {printf("g");}
{TK_H}   {printf("h");}
{TK_I}   {printf("i");}
{TK_J}   {printf("j");}
{TK_K}   {printf("k");}
{TK_L}   {printf("l");}
{TK_M}   {printf("m");}
{TK_N}   {printf("n");}
{TK_O}   {printf("o");}
{TK_P}   {printf("p");}
{TK_Q}   {printf("q");}
{TK_R}   {printf("r");}
{TK_S}   {printf("s");}
{TK_T}   {printf("t");}
{TK_U}   {printf("u");}
{TK_V}   {printf("v");}
{TK_W}   {printf("w");}
{TK_X}   {printf("x");}
{TK_Y}   {printf("y");}
{TK_Z}   {printf("z");}
.        ;
```

E por fim, na parte de funções auxiliares, foi criada a função main, onde é carregado o arquivo.txt e executado a decodificação. Veja o código a seguir:

```
int main(){
    extern FILE *yyin;
    yyin = fopen("arquivo.txt", "r");
    yylex();
}
```

Link para o repositório no github com o código-fonte:

<https://github.com/Romulo-Menezes/Compiladores>