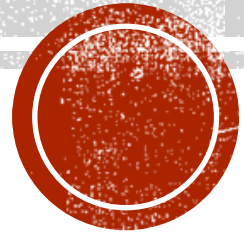
APS
LINGUAGEM CRIPTOGRAFADA POR CIFRA DE CÉSAR

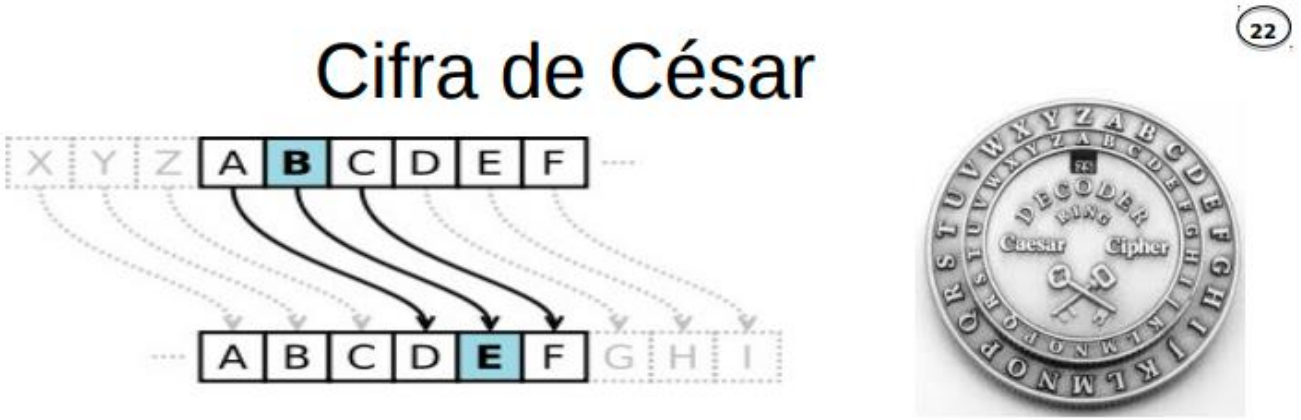
WILLIAM SILVA

LÓGICA DA COMPUTAÇÃO 2022/1



MOTIVAÇÃO

- Conexão de aprendizados: Tecnologias Hacker e Lógica da Computação
- Contexto: início das aulas de Criptografia



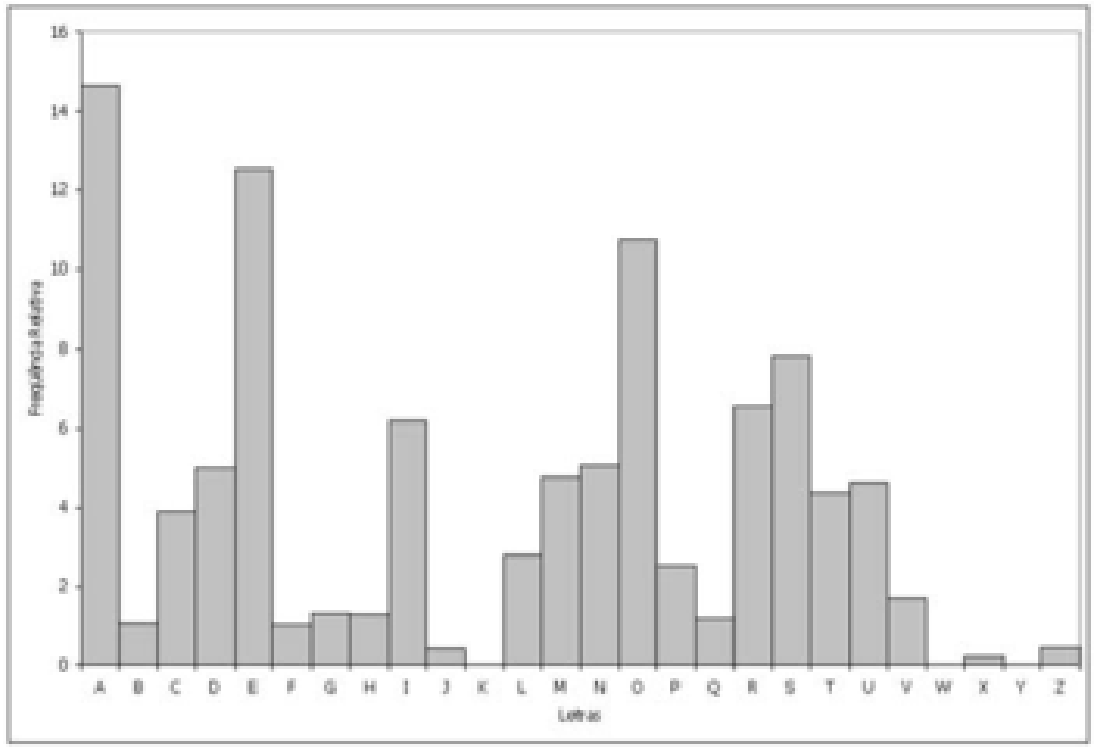
- Aula do professor Rodolfo Avelino ao lado
- Ademais, pensei na ideia de criptografar um programa de computador, resultando a mensagem real apenas para quem tem o compilador oficial. É a mesma ideia das chaves para descriptografar, porém de um jeito muito mais simples.

CARACTERÍSTICAS DA CIFRA DE CÉSAR

- É uma técnica de criptografia bem simples, na qual cada letra é substituída por outra. Por exemplo, caso o A seja substituído por G, o B seria pelo H, C pelo I, e assim por diante.
- Nome porque Júlio César, um político romano, utilizava essa estratégia para se comunicar com seus militares
- A minha cifragem utilizada:
  - Normal: ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789
  - Cifrado: HIJKLMNOPQRSTUVWXYZABCDEFG 3456789012
- Para quem tem conhecimento, é fácil de decifrar. Para quem não tem, se torna bem ilegível.

CURIOSIDADES

- A descriptografia é realizada, por quem não tem conhecimento da técnica utilizada, por meio de técnicas, como a de Análise de Frequência
- Por exemplo, na Língua Portuguesa, a frequência usada nas palavras de cada letra é:



- Ou seja, se num texto criptografado de césar tiver uma frequência de 14% no texto, ou algo próximo, supõe-se que seja o A
- Assim, vai se decifrando cada letra para conseguir o texto original

CARACTERÍSTICAS DA MINHA LINGUAGEM

- Linguagem C criptografada
  - Função principal: main (thpu)
- Diferenças
  - Operações Aritméticas
    - Normal: a (+ | - | * | /) b
    - Transição: a (plus | minus | mult | div) b
    - Cifrada: a (wsbz | tpubz | tbsa | kpc) b
  - Operações Lógicas
    - Normal: a (< | > | && | || | !) b
    - Transição: a (lt | gt | and | or | not) b
    - Cifrada: a (sa| na| huk | vy | uva) b
- Arquivo de extensão '.cr'
- Identifiers não podem começar com h ou j, porque eles representam { e }, respectivamente.

CARACTERÍSTICAS DA MINHA LINGUAGEM

Tokens que se mantiveram

- ,
- (
-)
- .
- ;

Tokens que mudaram

- if: pm
- else: lszl
- while: dopsl
- scanf: zjhum
- ==: pz (vem de “is”)
- =: lxbhs
- str, int, void: zay, pua, cvpk

EXEMPLO 1

Linguagem C

```c
int main() {
    printf(2);
}
```

Cifra de César

```
pua thpu() h
    wypuam(5);
j
```

Saída de Ambas

```
2
```

EXEMPLO 2

Linguagem C (do compilador)

```c
int main() {
    int a;
    str b;
    a = 1;
    b = "hello";

    printf(a);
    printf(a . b);
    printf(b . a);
}
```

Cifra de César

```
pua thpu() h
    pua a;
    zay b;
    a lxbhs 4;
    b lxbhs "olssv";

    wypuam(a);
    wypuam(a . b);
    wypuam(b . a);
j
```

Saída de Ambas

```
1
1hello
hello1
```

EXEMPLO 3

Linguagem C (do compilador)

```c
int main() {
    int a;
    int b;
    str c;
    a = 5;
    b = 0;
    c = "hello";

    if (a == b) {
        printf(c);
    } else {
        printf(a + b);
        printf(5 * 9 + a);
    }

    if (a == b)
        printf(c);
    else
        printf(a + b);

    printf(a);
    printf(b);
    printf(c);
}
```

Cifra de César

```
pua thpu() h
    pua a;
    pua b;
    zay c;
    a lxbhs 8;
    b lxbhs 3;
    c lxbhs "olssv";

    pm (a pz b) h
        wypuam(c);
    j lszl h
        wypuam(a wsbz b);
        wypuam(8 tbsa 2 wsbz a);
    j

    pm (a pz b)
        wypuam(c);
    lszl
        wypuam(a wsbz b);


    wypuam(a);
    wypuam(b);
    wypuam(c);
j
```

Saída de Ambas

```
5
50
5
5
0
hello
```

EXEMPLO 4

Linguagem C (do compilador)

```c
int soma(int x, int y) {
  int a;
  a = x + y;
  printf(a);
  return(a);
}

int main() {
  int a;
  int b;
  a = 3;
  b = 2;
  b = soma(a, 4);
  printf(a);
  printf(b);
}
```

Cifra de César

```
pua soma(pua x, pua y) h
  pua a;
  a lxbhs x wsbz y;
  wypuam(a);
  ylabyu(a);
j

pua thpu() h
  pua a;
  pua b;
  a lxbhs 6;
  b lxbhs 5;
  b lxbhs soma(a, 7);
  wypuam(a);
  wypuam(b);
j
```

Saída de Ambas

```
7
3
7
```

EXEMPLO 5

Linguagem C (do compilador)

```c
int sub(int a, int b) {
    int x;
    x = a - b;
    return (x);
}

str concat(str a, str b) {
    str x;
    x = a . b;
    return (x);
}
```

```c
int main() {
    int a;
    int b;
    str c;
    str d;

    a = 5;
    b = 1910;

    printf(a);
    printf(b);

    c = "Essa é uma mensagem super secreta. Vc não deveria compartilhar com ninguém:";
    d = " VAI CORINTHIANS ";
    printf(sub(a, b));
    printf(concat(c, d));
    printf(d . b);
    /*
        ALGUM COMENTARIO
    */
}
```

Cifra de César

```
pua zbi(pua a, pua b) h
    pua x;
    x lxbhs a tpubz b;
    ylabyu (x);
j

zay concat(zay a, zay b) h
    zay x;
    x lxbhs a . b;
    ylabyu (x);

j       pua thpu() h
    pua a;
    pua b;
    zay c;
    zay d;

    a lxbhs 8;
    b lxbhs 4243;

    wypuam(a);
    wypuam(b);

    c lxbhs "Lzzh l bth tluzhnlt zbwly zljylah cj uhv klclyph jvtwhyapsohy jvt upunblt";
    d lxbhs " CHP JVYPUAOPHUZ ";
    wypuam(zbi(a, b));
    wypuam(concat(c, d));
    wypuam(d . b);

    /*
        HSNBT JVTLUAHYPV
    */
j
```

Saída de Ambas

```
5
1910
-1905
Essa e uma mensagem secreta você não deveria compartilhar com ninguém VAI CORINTHIANS
VAI CORINTHIANS 1910
```

EXEMPLO 6

Linguagem C (do compilador)

```c
int main(){
    int a;
    a = 1;
    while (a < 10) {
        printf(a);
        a = a + 1;
    }
}
```

Cifra de César

```
pua thpu() h
    pua a;
    a lxbhs 4;

    dopsl (a sa 43)h
        wypuam(a);
        a lxbhs a wsbz 4;
    j
j
```

Saída de Ambas

1
2
3
4
5
6
7
8
9

COMO TESTAR

- git clone https://github.com/williamars/my-programming-language
- python main.py example/example-01.cr
- Podendo ir até o example-06.cr

REFERÊNCIAS

- Minha implementação:
  - *https://github.com/williamars/my-programming-language*
- *Ótima referência de criptografia e descriptografia*
  - https://marciapsilva.github.io/cifra-de-cesar/