**Exercícios para Entrega 31/08/18**

1. O código de Cesar é uma das técnicas de criptografia mais simples e conhecidas. É um tipo de substituição no qual cada letra do texto é substituída por outra, que se apresenta n posições após ela no alfabeto. Por exemplo, com uma troca de três posições a letra A seria substituída por D, B se tornaria E e assim por diante. Escreva um programa que faça uso da função void cesar(char[], int posicoes) . Entre com uma string, gere uma nova string e imprima uma string codificada. Exemplo:

**String**: a ligeira raposa marrom saltou sobre o cachorro cansado

**Nova String**: d oljhlud udsrvd pduurp vdowrx vreuh r fdfkruur fdqvdgr

1. Crie uma função que recebe uma string e transforma alguns dos caracteres em maiúsculos e outros em minúsculos. Faça sorteios com [a **função rand()** para gerar números aleatórios em C](http://www.cprogressivo.net/2013/03/Como-gerar-numeros-aleatorios-em-C-com-a-rand-srand-e-seed.html), que serão usados para escolher os índices dos caracteres que serão alterados. Como parâmetro deve ser recebida a quantidade de sorteios que devem ser executados

void transforma(char [], int qtde\_sorteios);

1. Crie uma função que recebe uma string e um caractere, e apague todas as ocorrências desses caractere na string.
2. Crie uma função que retira todas os caracteres repetidos de uma string.
3. Faça uma função chamada compara() que recebe duas strings (s1 e s2) como parâmetros e compara os seus conteúdos. A função deve retornar a posição onde s2 começa dentro de s1 (caso s2 faça parte de s1) e -1, caso contrário. Repare que s2 pode estar contida em s2 em qualquer posição. Não use a função strstr(). **Exemplo**: se a função recebesse as strings s1 à “atenciosamente” e s2 à “cio”, a função deveria retornar 4.

**DICA:**

A função que gera números aleatórios em C é a *rand()*.

Ela gera números entre 0 e RAND\_MAX*,*onde esse RAND\_MAX é um valor que pode variar de máquina pra máquina.

Pra usar a função *rand()*, temos que adicionar a biblioteca *time.h* e para saber o valor de RAND\_MAX, temos que usar a função *stdlib.h*.

Veja um programa que gera 10 números aleatórios em C:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

**int** **main**(void)

{

**int** i;

**printf**("intervalo da rand: [0,%d]\n", RAND\_MAX);

**for**(i=1 ; i <= 10 ; i++)

**printf**("Numero %d: %d\n",i, **rand**());

}

|  |  |
| --- | --- |
|  |  |

**Alimentando a rand() com a srand() : seed**

Parece tudo ok, não? Veja os números gerados no exemplo passado.

Agora feche e rode o programa novamente. E de novo, de novo...notou?  
(não vale enrolar! teste mesmo!)

A seqüência gerada é sempre a mesma. Para mudar isso, vamos alimentar a função *rand()* com uma semente, com um número, que é o tempo atual.

Assim, toda vez que rodarmos o programa, a *rand()* pega um número de tempo diferente e gera uma seqüencia diferente.

Para fazer isso, basta usar a função *srand()*, que será responsável por alimentar a *rand().*Fazemos isso adicionando o comando: *srand( (unsigned)time(NULL) );*antes da *rand()*.

E agora nosso aplicativo em C gera uma seqüencia diferente toda vez que rodamos:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

**int** **main**()

{

**int** i;

**printf**("intervalo da rand: [0,%d]\n", RAND\_MAX);

**srand**( (unsigned)**time**(**NULL**) );

**for**(i=1 ; i <= 10 ; i++)

**printf**("Numero %d: %d\n",i, **rand**());

}

**Escolhendo uma faixa de números aleatórios em C**

Ok, agora você é mestre nas artes de geração aleatória de números em linguagem C.

Porém, seu código gera números entre 0 e RAND\_MAX.

E se você quiser gerar entre 0 e 10? Ou entre 1 e mil?

Ou entre 0.00 e 1.00?

Para escolher a faixa de valores vamos usar operações matemáticas, principalmente o operador de módulo, também conhecido como resto da divisão: %

Para fazer com que um número ‘*x*’ receba um valor entre 0 e 9, fazemos:

x = rand() % 10

Para fazer com que um número ‘*x*’ receba um valor entre 1 e 10, fazemos:

x = 1 + ( rand() % 10 )

Para fazer com que um número ‘*x*’ receba um valor entre 0.00 e 1.00, primeiro geramos números inteiros, entre 0 e 100:

x = rand() % 101

Para ter os valores decimais, dividimos por 100:

x = x/100;