**Sistema**

Limpar o terminal: system ("clear");

**Operadores**

Adição: +

Subtração: -

Multiplicação: \*

Divisão: /

Resto da divisão: %

Maior que: >

Maior ou igual que: >=

Menor que: <

Menor ou igual que: <=

Igual a: ==

Diferente de: !=

E: &&

Ou: ||

Vetor: variável[variável]

Matriz: variável[variável][variável]

**Laços**

Fazer enquanto: do { o que será feito } while (variável enquanto é executado)

Durante: for (valor inicial da variável; variável enquanto é executado; a cada vez que for executado) { o que será feito }

**Estrutura**

Base da estrutura: typedef struct { tipos de variáveis e vetor } nome da estrutura;

Declarar: nome da estrutura e variável

**Biblioteca: #include <stdio.h>**

Números inteiros: %i :int:

Receber números com ponto: %f :float:

Exibir números com ponto: %.1f :float:

Letra: %c :char:

Palavra: %s :char:

Exibir texto: printf (“texto”);

Receber conteúdo do usuário: scanf (“%tipo de variável”, &);

Receber conteúdo do usuário: gets (variável);

Pular linha: \n

Pular espaço: \t

Se: if (variáveis) { o que será feito }

Se não se: else if (variáveis) { o que será feito }

Se não: else (variáveis) { O que será feito }

**Arquivos**

FILE: \*variável;

Abrir arquivo .txt: variável FILE = fopen (“nome do arquivo.txt”, “modo de abertura”);

Lê caractere por caractere: fgetc (variável, variável FILE)

Lê toda a linha: fgets (variável, número de caracteres, variável FILE)

Escreve caractere por caractere: fputc (variável, variável FILE)

Escreve todos os caracteres: fputs (variável, variável FILE)

Renomear: rename (“nome do arquivo.txt”, “novo nome do arquivo.txt”);

Remover: remove (“nome do arquivo.txt”);

Variável final do arquivo .txt: EOF ou NULL

Fechar arquivo .txt: fclose (variável FILE);

Colocar ao final de cada linha: \n

FILE: \*variável;

Abrir arquivo .bin: variável FILE = fopen (“nome do arquivo.bin”, “modo de abertura”);

Lê as estruturas: fread (variável estrutura, sizeof (tipo de estrutura), número de estruturas, variável FILE);

Escreve as estruturas: fwrite (variável estrutura, sizeof (tipo de estrutura), número de estrutura, variável FILE);

Renomear: rename (“nome do arquivo.bin”, “novo nome do arquivo.bin”);

Remover: remove (“nome do arquivo.bin”);

Fechar arquivo .bin: fclose (variável FILE);

**Modos De Abertura**

Abre um arquivo texto já existente para leitura: r

Abre um arquivo texto já existente para leitura e escrita: r+

Abre ou cria um arquivo texto para escrita apagando seu conteúdo: w

Abre ou cria um arquivo texto para escrita e leitura apagando seu conteúdo: w+

Abre ou cria um arquivo texto para escrita e leitura mantendo seu conteúdo - ponteiro é colocado no início do arquivo para leitura e no final para escrita: a+

Modo recomendado: a+

**Cores**

#define preto printf (“\33[30m”);

#define vermelho printf (“\33[31m”);

#define verde printf (“\33[32m”);

#define amarelo printf (“\33[33m”);

#define azulescuro printf (“\33[34m”);

#define roxo printf (“\33[35m”);

#define azulclaro printf (“\33[36m”);

#define branco printf (“\33[37m”);

Trocar as: “

**Biblioteca: #include <stdio\_ext.h>**

Limpar o espaço de memória do Linux: \_\_fpurge(stdin);

Limpar o espaço de memória do Windows: fflush(stdin);

**Biblioteca: #include <math.h>**

Receber números com ponto: %lf :double:

Exibir números com ponto: %.1lf :double:

Potência: variável = pow (variável, variável) :float:

Raiz: variável = sqrt (pow (variável, variável))

Hipotenusa: variável = sqrt (pow (variável, 2) + pow (variável, 2))

Necessário para executar no terminal: -lm

**Biblioteca: #include <stdlib.h>**

Alocar uma quantidade de memória vetor: \*variável = malloc (TAM \* sizeof (tipo da variável));

Alocar uma quantidade de memória matriz: \*\*variável[variável] = malloc (TAM \* sizeof (tipo da variável\*));

**Biblioteca: #include <time.h>**

Números aleatórios: srand(time(NULL));

variavel = número inicial + rand() % números acima do inicial

**Biblioteca: #include <string.h>**

Quantidade de caracteres: variável = strlen (variável);

Comparar os caracterespor: variável = strcmp (variável, variável);

Comparar a quantidade de caracteres escolhido: variável = strncmp (variável, variável, TAM);

Copiar os caracteres: variável = strcpy (variável, variável);

Copiar a quantidade de caracteres escolhido: variável = strncpy (variável, variável, TAM);

Copiar os caracteres sem apagar: strcat (variável, variável);

Copiar a quantidade de caracteres escolhido sem apagar: strncat = (variável, variável, TAM);