Para todas as questões, compile-as com o gcc e execute-as via terminal.

1. Crie um código em C para escrever "Ola mundo!" em um arquivo chamado 'ola\_mundo.txt'.

#include <stdio.h>

#include <stdlib.h>

int main(){

FILE \*fp;

fp = fopen("ola\_mundo.txt","w");

fputs("Ola mundo!",fp);

fclose(fp);

return 0;

}

1. Crie um código em C que pergunta ao usuário seu nome e sua idade, e escreve este conteúdo em um arquivo com o seu nome e extensão '.txt'. Por exemplo, considerando que o código criado recebeu o nome de 'ola\_usuario\_1':

$ ./ola\_usuario\_1

$ Digite o seu nome: Eu

$ Digite a sua idade: 30

$ cat Eu.txt

$ Nome: Eu

$ Idade: 30 anos

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main(){

FILE \*fp;

char nome\_arq[30],nome[30], idade[5];

printf("Digite o seu nome: ");

gets(nome);

printf("Digite a sua idade: ");

gets(idade);

strcpy(nome\_arq,nome);

strcat(nome\_arq,".txt");

fp = fopen(nome\_arq,"w");

fputs("Nome: ",fp);

fputs(nome,fp);

fputs("\n",fp);

fputs("Idade: ",fp);

fputs(idade,fp);

fputs(" anos\n",fp);

fclose(fp);

return 0;

}

1. Crie um código em C que recebe o nome do usuário e e sua idade como argumentos de entrada (usando as variáveis argc e \*argv[]), e escreve este conteúdo em um arquivo com o seu nome e extensão '.txt'. Por exemplo, considerando que o código criado recebeu o nome de 'ola\_usuario\_2':

$ ./ola\_usuario\_2 Eu 30

$ cat Eu.txt

$ Nome: Eu

$ Idade: 30 anos

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main(int argc,char \*\*argv){

FILE \*fp;

char \*nome\_arq,\*nome, \*idade;

nome = argv[1];

idade = argv[2];

strcpy(nome\_arq,nome);

strcat(nome\_arq,".txt");

fp = fopen(nome\_arq,"w");

fputs("Nome: ",fp);

fputs(nome,fp);

fputs("\n",fp);

fputs("Idade: ",fp);

fputs(idade,fp);

fputs(" anos\n",fp);

fclose(fp);

return 0;

}

1. Crie uma função que retorna o tamanho de um arquivo, usando o seguinte protótipo: int tam\_arq\_texto(char \*nome\_arquivo); Salve esta função em um arquivo separado chamado 'bib\_arqs.c'. Salve o protótipo em um arquivo chamado 'bib\_arqs.h'. Compile 'bib\_arqs.c' para gerar o objeto 'bib\_arqs.o'.

int tam\_arq\_texto(char \*nome\_arquivo);

#include <stdio.h>

int tam\_arq\_texto(char \*nome\_arquivo){

FILE \*fp;

int size;

fp = fopen(nome\_arquivo,"r");

fseek(fp, 0, SEEK\_END);

size = ftell(fp);

fclose(fp);

return size;

}

1. Crie uma função que lê o conteúdo de um arquivo-texto e o guarda em uma string, usando o seguinte protótipo: void le\_arq\_texto(char \*nome\_arquivo, char \*conteúdo); Repare que o conteúdo do arquivo é armazenado no vetor conteudo[]. Ou seja, ele é passado por referência. Salve esta função no mesmo arquivo da questão 4, chamado 'bib\_arqs.c'. Salve o protótipo no arquivo 'bib\_arqs.h'. Compile 'bib\_arqs.c' novamente para gerar o objeto 'bib\_arqs.o'.

int tam\_arq\_texto(char \*nome\_arquivo);

void le\_arq\_texto(char \*nome\_arquivo, char \*conteúdo);

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int tam\_arq\_texto(char \*nome\_arquivo){

FILE \*fp;

int size;

fp = fopen(nome\_arquivo,"r");

fseek(fp, 0, SEEK\_END);

size = ftell(fp);

fclose(fp);

return size;

}

void le\_arq\_texto(char \*nome\_arquivo, char \*conteudo){

FILE \*fp;

int size,i=0;

char c;

fp = fopen(nome\_arquivo,"r");

while((c = getc(fp))!=EOF){

conteudo[i] = c;

i++;

}

conteudo[i]='\0';

fclose(fp);

}

1. Crie um código em C que copia a funcionalidade básica do comando cat: escrever o conteúdo de um arquivo-texto no terminal. Reaproveite as funções já criadas nas questões anteriores. Por exemplo, considerando que o código criado recebeu o nome de 'cat\_falsificado':

$ echo Ola mundo cruel! Ola universo ingrato! > ola.txt

$ ./cat\_falsificado ola.txt

$ Ola mundo cruel! Ola universo ingrato!

#include <stdio.h>

#include <stdlib.h>

#include "bib\_arqs.h"

int main(int argc,char \*\*argv){

char \*a;

int size;

size = tam\_arq\_texto(argv[1]);

a = malloc(size+1);

le\_arq\_texto(argv[1],a);

printf("%s",a);

return 0;

}

1. Crie um código em C que conta a ocorrência de uma palavra-chave em um arquivo-texto, e escreve o resultado no terminal. Reaproveite as funções já criadas nas questões anteriores. Por exemplo, considerando que o código criado recebeu o nome de 'busca\_e\_conta':

$ echo Ola mundo cruel! Ola universo ingrato! > ola.txt

$ ./busca\_e\_conta Ola ola.txt

$ 'Ola' ocorre 2 vezes no arquivo 'ola.txt'.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "bib\_arqs.h"

int countOccurrences(char \* str, char \* toSearch);

int main(int argc,char \*\*argv){

char \*str\_file;

int size,count;

size = tam\_arq\_texto(argv[2]);

str\_file = malloc(size+1);

le\_arq\_texto(argv[2],str\_file);

count = countOccurrences(str\_file,argv[1]);

printf("'%s' ocorre %d vezes no arquivo '%s'\n",argv[1],count,argv[2]);

return 0;

}

int countOccurrences(char \* str, char \* toSearch)

{

int i, j, found, count;

int stringLen, searchLen;

stringLen = strlen(str); // length of string

searchLen = strlen(toSearch); // length of word to be searched

count = 0;

for(i=0; i <= stringLen-searchLen; i++)

{

/\* Match word with string \*/

found = 1;

for(j=0; j<searchLen; j++)

{

if(str[i + j] != toSearch[j])

{

found = 0;

break;

}

}

if(found == 1)

{

count++;

}

}

return count;

}