

SSC0600 – Introdução à Ciência de Computação I – 2025 (ICC1 - Teoria)

Professor responsável: *Fernando Santos Osório*

Semestre: 2025/1

Horário: Terça 10h – 13h (Prova no Lab.)

Wiki: SSC-600-2025(fosorio)

Web: <http://www.icmc.usp.br/~fosorio/>

DATA DA PROVA: 24 / 06 / 2025 - P2 Final

NRO. USP: < Colocar o seu NUSP no **programa fonte** > COMO COMENTÁRIO no CÓDIGO

NOME : < Colocar o seu Nome no **programa fonte** > COMO COMENTÁRIO no CÓDIGO

>> COLOCAR SEU NOME E NRO. USP COMO COMENTÁRIO DO PROGRAMA FONTE ENTREGUE!

PROVA FINAL – SSC0600 - ICC 1

QUESTÃO ÚNICA – Implementar um programa que lê um arquivo texto para uma estrutura de dados em memória (struct com alocação dinâmica e em tempo de execução), armazenando as linhas do texto em memória e tendo um campo que permite enumerar as linhas do arquivo (campos número da linha e texto da linha). O usuário indicará como será a numeração (início e incremento do número das linhas). Uma vez lido o texto, gerada a numeração das linhas conforme indicado pelo usuário (entrada de teclado), o texto deve ser exibido na tela e salvo em disco em um formato que exibe o número da linha e o texto de cada linha. Por fim, o programa deve realizar uma estatística a respeito de algumas palavras-chaves, contando o número de ocorrências destas palavras-chave no texto e exibindo na tela a contagem. As palavras-chave buscadas serão “int “, “char “, “float “ e “double “ (palavras sempre terminadas com um espaço em branco após a palavra).

O programa deve usar os conceitos vistos em aula (structs, alocação dinâmica de memória, sub-rotinas com parâmetros por valor e por referência e arquivo texto), conforme especificado mais detalhadamente a seguir.

Implemente o seguinte programa descrito abaixo de acordo com o especificado.

O programa deve ser enviado para o RunCodes.ICMC da Disciplina SSC0600 (Prova P2 Final)

RunCodes SSC0600 - <https://runcodes.icmc.usp.br/offerings/view/96> (SSC0600 - ICC 1 Prof. FOsorio)

Descrição do Programa:

Implemente um programa que realize as seguintes tarefas (sub-rotinas):

- Ler o arquivo e contar o número de linhas de texto que este arquivo contém (abre, lê, conta e “descarta” o conteúdo, fecha o arquivo), retornando o número de linha total que o arquivo possui. O arquivo **termina** com uma linha que começa com o caracter ‘@’ (ver exemplo a seguir) – e define o fim da leitura do arquivo.
- Ler TODO o arquivo para a memória, pois agora, como sabemos o número de linhas, do arquivo podemos alocar uma estrutura (struct) com espaço para armazenar **todos** os dados do arquivo. Portanto, a leitura do arquivo consiste em criar (calloc) uma struct com campos para o número da linha (int) e para o texto da linha lida (string de tamanho máximo 150), do tamanho suficiente para guardar todas as linhas do arquivo (abre novamente o arquivo, lê e armazena na memória). Em caso de erro na leitura do arquivo ou na alocação de memória, exibir na tela a mensagem “ERRO\n” (sim, com ‘\n’!) e abortar.

- Ler do teclado dois inteiros, o primeiro indica o número inicial da numeração das linhas do arquivo, e o segundo número indica o incremento (passo) de avanço na numeração das linhas do arquivo. Executar uma sub-rotina que recebe estes valores (início e incremento) e ajusta a numeração das linhas nos dados em memória. Supondo que os valores lidos sejam 1 e 1, isso significa que vamos numerar as linhas começando na linha 1 e seguindo de 1 em 1 a contagem das linhas (ver exemplo abaixo). Se forem lidos os valores 10 e 10, isso significa que vamos começar a numeração em 10 seguindo de 10 em 10 a contagem das linhas (ver segundo exemplo abaixo). Exemplo - Arquivo texto de Entrada e Saídas:

Nome do Arquivo de Entrada: “texto.txt” (lido do disco)

Nome do Arquivo de Saída: “numerado.txt” (gravado no disco)

Arquivo de Entrada:

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    printf("Hello world! \n");
    return 0;
}
```

Arquivo Numerado: início 1, incremento 1

```
000001 #include <stdio.h>
000002 #include <stdlib.h>
000003
000004 int main(void)
000005 {
000006     printf("Hello world! \n");
000007     return 0;
000008 }
```

Arquivo Numerado: início 10, incremento 10 (saída na tela e em disco)

```
000010 #include <stdio.h>
000020 #include <stdlib.h>
000030
000040 int main(void)
000050 {
000060     printf("Hello world! \n");
000070     return 0;
000080 }
```

- Faça uma rotina que exiba na tela a listagem EXATAMENTE como apresentado no exemplo acima, usando a formatação “%06d” para exibir o número da linha com 6 casas, alinhado à direita e começando com zeros. Liste o arquivo na tela com as linhas numeradas, e também salve em disco um arquivo texto neste mesmo formato, com linhas numeradas e como o nome “numerado.txt”.

- Por fim, implemente uma rotina que realize a estatística das palavras-chaves (“int “, “char “, “float “ e “double “). Esta rotina deve buscar em todas as linhas do arquivo armazenado em memória pelas strings das palavras-chave, implementando contadores do número de linhas onde foi encontrado “int “, número de linhas onde foi encontrado “char “, e linhas com “float “ e com “double “. Esta rotina deve OBRIGATORIAMENTE retornar 4 valores (por referência) que são os contadores do número de ocorrências de cada uma das palavras-chave. **Use string.h e a função strstr !**

- O programa principal (main) deve, portanto, executar as seguintes tarefas:

1. Estimar o número de linhas do arquivo texto de entrada.
2. Alocar dinamicamente uma estrutura para armazenar todo o arquivo e ler o arquivo para a memória.
3. Ler do teclado os valores “início” e “incremento” e renumerar as linhas do arquivo.
4. Exibir na tela a listagem do arquivo com as linhas numeradas.
5. Salvar em disco a mesma listagem exibida na tela (arquivo de entrada com suas linhas numeradas).
6. Executar a estatística das palavras-chave retornando a contagem ao programa principal.
7. Exibir na tela as estatísticas da contagem de palavras-chave, conforme o exemplo abaixo.

Exemplo de Saída: (para o arquivo texto.txt com o programa Hello World!) e entradas: 1 1

```
000001 #include <stdio.h>
000002 #include <stdlib.h>
000003
000004 int main(void)
000005 {
000006     printf("Hello world! \n");
000007     return 0;
000008 }
NI:1
NC:0
NF:0
ND:0
```

NI é o número de “int”, NC é o número de “char”, NF é o número de “float” e ND é o número de “double”. Todos são exibidos na forma indicada acima, com ‘\n’ após cada valor exibido.

LINK DOS ARQUIVOS DA PROVA:

<https://drive.google.com/drive/folders/1ID4odw-GOxKjizSdXKHkfaoZRLmntfYO?usp=sharing>

LINK CURTO: <https://tinyurl.com/zz3ssatx>

(Links disponíveis também no RUNCODES - Detalhes/Descrição da Prova)

(*) Prova inspirada no comando NL do Linux => nl -b a -n rz -v10 -i10 texto.txt

<https://www.geeksforgeeks.org/linux-unix/nl-command-in-linux-with-examples/>

BOA PROVA!!!

REGRAS EM RELAÇÃO REALIZAÇÃO DESTA PROVA

1. A PROVA É INDIVIDUAL com consulta Papel (Livros, papel impresso ou escrito) e formato Digital (Internet, Wiki, Pendrive), porém SEM CONSULTAR ou SE COMUNICAR COM HUMANOS ou QUALQUER OUTRA FORMA DE VIDA TERRESTRE ou EXTRA-TERRESTRE!
Não podem usar de formas de comunicação com pessoas externas, além do professor, referente a prova, seja por celular (manter desligado/guardado), por e-mail, por whatsapp, por mensagens ou fóruns (“ao vivo”), com colegas, etc.
NÃO É PERMITIDO O EMPRÉSTIMO DE MATERIAL (Cadernos, Anotações, Livros, etc).
2. A PROVA DEVE SER REALIZADA USANDO OS COMPUTADORES DO LABORATÓRIO.
Não será permitido o uso de Notebooks, Tablets ou Celulares de uso pessoal.
3. A PROVA DEVE SER ENTREGUE via RunCodes.icmc (pode submeter várias vezes)
Entrega é via <https://runcodes.icmc.usp.br/> Disciplina SSC0600 – Prova P2 FINAL
4. EM CASO DE PROBLEMAS GRAVES no RunCodes.icmc pode ser entregue por E-Mail para: fosorio@icmc.usp.br com cópia (Cc:) para fosorio@gmail.com => PROFESSOR DEVE SER AVISADO!
Com assunto/subject: Prova-P2 <Seu_Nome> <Nro_USP>
**ANEXANDO O PROGRAMA FONTE (“.C” e o Project “.CBP”) – NÃO ANEXAR EXECUTÁVEIS!
NÃO ANEXAR OBJ, BIN e EXE!!! Seu e-mail não será recebido se for anexado bin, obj ou exe!**
5. A PROVA USUALMENTE DEVE SER REALIZADA USANDO O CODEBLOCKS DO LABORATÓRIO, e enviada no RUNCODES, se for usar outro Compilador/IDE, INDIQUE NO E-MAIL qual AMBIENTE, COMPILADOR e IDE usou!
6. AO ENTREGAR A PROVA NO RUNCODES ou POR E-MAIL O(A) ALUNO(A) CONCORDA COM ESTAS REGRAS E SE COMPROMETE A FAZER A PROVA INDIVIDUALMENTE!