

Faça um programa, em C, que leia um vetor de inteiros, ordene o vetor em ordem crescente e imprima o vetor em um arquivo. O vetor deve ser impresso no formato de inicialização de vetores de C. O tamanho do vetor pode ser fixo em 10 posições.

Seu programa deve implementar e chamar as seguintes funções auxiliares:

```
void inicializa_vetor(int n, int *vet);
```

A função *inicializa_vetor* deve zerar todas as n posições de *vet*.

```
int le_vetor(int n, int *vet);
```

Deve abrir o arquivo “entrada.txt” para leitura e armazenar cada linha em uma posição do vetor *vet*. Repare que cada linha do arquivo corresponde a um número inteiro. O arquivo deve parar de ser lido quando chegar ao final ou se o número de linhas ultrapassar o número de posições em *vet*. A função *le_vetor* deve retornar 0 se não conseguir abrir o arquivo “entrada.txt”, -1 se o arquivo for vazio ou 1 se a leitura foi efetuada com sucesso.

```
int imprime_vetor(int n, int *vet);
```

Deve abrir o arquivo “saida.txt” para escrita e imprimir no arquivo o conteúdo do vetor *vet*, no formato de inicialização de vetores em C. A função *imprime_vetor* deve retornar 0 se não conseguir abrir o arquivo “saida.txt” ou 1 se a escrita foi efetuada com sucesso.

```
void ordena_vetor(int n, int *vet);
```

A função *ordena_vetor* deve ordenar o vetor *vet* em ordem crescente.

A *main* do seu programa deve fazer uso das funções auxiliares implementadas. Lembre que o seu programa deve zerar o vetor declarado, inicializa-lo com as informações lidas do arquivo “entrada.txt”, ordena-lo e em seguida imprimi-lo no arquivo “saida.txt”. O programa deve abortar a execução com uma mensagem de erro no seguintes casos:

- “Não foi possível abrir o arquivo entrada.txt!”, *le_vetor* retorna 0;
- “Arquivo entrada.txt está vazio!”, *le_vetor* retorna -1;
- “Não foi possível abrir o arquivo saida.txt!”, *imprime_vetor* retorna 0;

Exemplo do arquivo "entrada.txt"

2
5
6
1
3
4
9
10
7
8

Exemplo do arquivo "saida.txt"

{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, }