



Exercícios de Ponteiros e Arquivos

1. Seja o seguinte trecho de programa:

```
int i = 3, j = 5;  
int *p, *q;  
p = &i;  
q = &j;
```

Qual é o valor das seguintes expressões?

a) $p = \&i$ b) $*p - *q$ c) $**\&p$ d) $3 * - *p / (*q) + 7$

2. Assumindo que `pulo[]` é um vetor do tipo `int`, quais das seguintes expressões referenciam o valor do terceiro elemento da matriz?

a) $*(pulo + 2)$ b) $*(pulo + 4)$ c) $pulo + 4$ d) $pulo + 2$

3. Responda as seguintes perguntas:

(a) Explique a diferença entre

a) `p++`; b) `(*p)++` c) `*(p++)`

4. Reescreva o programa abaixo usando ponteiros

```
int main ()  
{  
    float matrix [50][50];  
    int i, j;  
    for (i=0; i<50; i++)  
        for (j=0; j<50; j++)  
            matrix[i][j]=0.0;  
    return (0);  
}
```

5. Considerando as variáveis e ponteiros definidos abaixo; quais são as atribuições permitidas?

```
int i, *pi, **ppi;  
float f, *pf, **ppf;
```

- | | | |
|------------------|----------------|------------------|
| a) $i = f$ | e) $*pf = 10$ | i) $ppf = \&pf$ |
| b) $pf = \&i$ | f) $f = i$ | j) $**ppi = 100$ |
| c) $*pf = 5.9$ | g) $pi = \&f$ | |
| d) $*ppi = \&pi$ | h) $*pi = 7.3$ | |

6. Elabore um programa que declare 4 variáveis do tipo inteiro e 4 do tipo ponteiro de inteiro, e faça com que as variáveis ponteiro apontem para as variáveis inteiro. Coloque comandos que permitam modificar os valores das variáveis inteiras através das variáveis ponteiro.

7. Elabore um programa que armazene valores aleatórios em um vetor de inteiros de 10 posições e depois, em outro vetor de ponteiros de inteiros de tamanho 10, coloque os endereços dos valores do vetor de inteiros de forma ordenada crescente, ficando a primeira posição do vetor de ponteiros com o endereço do menor valor até a última posição que conterá o endereço do maior valor.
8. Faça um programa que preencha dois vetores *vet1* e *vet2*, cuja quantidade de elementos deverá ser informada, pelo teclado, pelo usuário, com números inteiros. Calcule e armazene em um vetor:
 - (a) a diferença entre *vet1* e *vet2*, isto é, todos os elementos de *vet1* que não existam em *vet2*, sem repetições; e
 - (b) a interseção entre *vet1* e *vet2*, isto é, apenas os elementos que aparecem nos dois vetores, sem repetições.

Apresente na tela os vetores originais e o vetor resultante para os itens a e b.

9. Escreva um programa que receba $n > 0$ números inteiros, armazene-os em um vetor e mostre como saída o maior e o menor elemento deste vetor. Seu programa deve conter uma função com a seguinte interface:

```
void min_max (int v [], int n, int *max, int *min)
```

10. Crie uma segunda função que encontrar o maior e o segundo maior elemento em um vetor *A* função deve ter a seguinte interface:

```
void dois_maiores (int v [], int n, int *p_maior, int *s_maior)
```

Que retorna os dois maiores elementos do vetor.

11. Escreva um programa que receba n números inteiros, com $n > 0$ par, calcule a soma e o produto deste conjunto através de uma função com a seguinte interface:

```
void soma_produto (int *a, int n, int *soma, int *produto)
```

O programa deve determinar quantos destes números são maiores do que a soma e quantos são maiores do que o produto. Observe que os números de entrada podem ser negativos.

12. Escreva uma função que leia um nome completo e retorne o sobrenome, ','(vírgula) e em seguida o restante do nome. Exemplo: Armando Luiz Nicolini Delgado ==> Delgado, Armando Luiz Nicolini. A função deve ter a seguinte interface:

```
char* SobrenomeVirgula(char* nome);
```

13. Escreva um programa que leia uma string e ordene seus caracteres em ordem crescente.
NOTA: Uma letra maiúscula vem antes de sua correspondente maiúscula, e ambas vem antes de sua letra sucessora. (Ex.: *A* vem antes de *a*, e ambas vem antes de *B* e *b*).
14. Suponha que temos dois arquivos cujo as linhas são ordenadas lexicograficamente. Por exemplo, estes arquivos podem conter nomes de pessoas, linha a linha, em ordem alfabética. Escreva um programa que receba, por linha de comando, os nomes destes

dois arquivos e de um terceiro. Crie este terceiro arquivo contendo todas as linhas destes dois arquivos ordenadas lexicograficamente.

Arquivo 1:

Antonio
Berenice
Diana
Solange
Sonia
Zuleica

Arquivo 2:

Carlos
Celia
Fabio
Henrique

Arquivo resultante:

Antonio
Berenice
Carlos
Celia
Diana
Fabio
Henrique
Solange
Sonia
Zuleica

15. Escreva um programa em C para contar a quantidade de palavras de um arquivo texto.
16. Escreva um programa para contar a quantidade de linhas de um arquivo texto.
17. Escreva um programa em C que gere e armazene em um arquivo uma matriz de tamanho $L \times C$, com os elementos da diagonal principal preenchidos com o valor '1' e o restante com o valor '0'.
18. Escreva um programa em C que abra um arquivo texto e que conte a quantidade de caracteres armazenados nele. Imprima o número na tela. O programa deve solicitar ao usuário que digite o nome do arquivo.
19. Escreva um programa em C que solicite ao usuário a digitação do nome de um arquivo texto já existente, e que então gere um outro arquivo, que será uma cópia do primeiro.
20. Considere um arquivo texto que armazene números em ponto flutuante em cada uma de suas linhas. Escreva um programa em C que determine o valor máximo, o valor mínimo e a média desses valores armazenados no arquivo. Imprima esses valores na tela.
21. Considere um arquivo texto que armazene caracteres variados, ou seja, um texto digitado. Escreva um programa que o leia e gere um novo arquivo que contenha somente as letras do arquivo original, na ordem em que lá aparecem (ou seja, caracteres de A-Z ou a-z).

22. Para um arquivo do mesmo tipo do da questão anterior (que armazene caracteres variados), escreva um programa em C que determine a média dos comprimentos de todas as palavras que se encontram nele. Entende-se por palavra um conjunto de caracteres de letras que está separado de outros conjuntos de caracteres no arquivo por um (ou mais) caractere de espaço em branco (' ', tabulações ou quebra de linha); e seu comprimento será a quantidade de caracteres que o formam.
23. Considere um arquivo texto como o usado na questão anterior. Faça um programa que o leia e que permita ao usuário consultar uma das linhas do arquivo, solicitando a ele que informe o índice n dessa linha. O programa deve imprimir a linha especificada ou a mensagem de que ela não existe.
24. Escreva um programa em C que receba via teclado o nome de um arquivo texto e uma palavra. O programa deve imprimir todas as linhas que possuem essa palavra.
25. Escreva um programa em C que receba via teclado o nome de um arquivo texto. O programa deve solicitar ao usuário que digite o índice da linha inicial e da linha final, e o programa deve imprimi-las e todas as linhas entre elas. Se o índice superior de linhas não existe, esse erro deve ser informado ao usuário (mas as linhas existentes devem, ainda, ser impressas).
26. Considere, agora, os números inteiros armazenados num arquivo texto como os usados nas questões anteriores. Escreva um programa em C que produza dois arquivos texto: o primeiro com os números pares da seqüência original e o segundo com os números ímpares. Os arquivos de saída devem conter um número por linha.