

01. [Damas, 2007] Escreva um programa em C que solicite ao usuário uma determinada data no formato *aaaa-mm-dd* e a mostre em seguida no formato *dd/mm/aaaa*. Não é necessário validar a data fornecida pelo usuário.
02. [Manzano, 1996] Escreva um programa em C que calcule a área de uma circunferência, através da fórmula $\text{área} = \text{pi} * \text{raio}^2$. Considere que o valor de *pi* seja 3.14159.
03. [Manzano, 1996] Escreva um programa em C que calcule a área de um triângulo, através da fórmula $\text{área} = (\text{base} * \text{altura}) / 2$.

04. [Damas, 2007] Escreva um programa que coloque na tela a seguinte frase:

```
Bem-vindos ao /Mundo\ da programação em "C"
```

05. [Damas, 2007] Escreva um programa que coloque na tela uma árvore com o seguinte formato:

```
      *
     * * *
    * * * * *
   / | \
```

06. [Manzano, 1996] Escreva um programa em C que leia uma temperatura em graus Centígrados e apresentá-la convertida em graus Fahrenheit. A fórmula de conversão é: $F = (9 * C + 160) / 5$, onde F é a temperatura em Fahrenheit e C é a temperatura em Centígrados.
07. [Mizrahi, 1990a] Escreva um programa em C que contenha uma única instrução e imprima na tela:

```
Está é a linha um.
Está é a linha dois.
```

08. [Damas, 2007] Escreva um programa que coloque na tela a seguinte saída:

```
Total      =      100%
IVA         =      17%
IRS         =      15%
-----
Líq.       =      68%
```

09. [Manzano, 1996] Escreva um programa em C que leia uma temperatura em graus Fahrenheit e apresentá-la convertida em graus Centígrados. A fórmula de conversão é: $C = (F - 32) * (5 / 9)$, onde F é a temperatura em Fahrenheit e C é a temperatura em Centígrados.
10. [Damas, 2007] Experimente a função `puts("Hello World");` (*put string*) para escrever a *string* Hello World e indique qual a diferença entre esta e a função `printf`. (*Nota: Essa função também faz parte do stdio.h*)
11. [Mizrahi, 1990a] Escreva um programa em C que declare três variáveis inteiras e atribua os valores 1, 2 e 3 a elas; declare três variáveis caracteres e atribua a elas as letras a, b e c; finalmente imprima na tela:

```
As variáveis inteiras contêm os números 1, 2 e 3.
As variáveis caracteres contêm os valores a, b e c.
```

12. [Manzano, 1996] Escreva um programa em C para calcular e apresentar o valor do volume de uma lata de óleo, utilizando a fórmula $\text{volume} = 3.14159 * \text{raio}^2 * \text{altura}$.

13. [Mizrahi, 1990a] Escreva um programa em C que converta a idade fornecida pelo usuário em anos para dias. Exemplo:

```
Digite sua idade em anos: 12.5
Sua idade em dias é 4562.
```

14. [Damas, 2007] Escreva um programa em C que apresente a seguinte saída:

```
1 -      Clientes
2 -      Fornecedores
3 -      Faturas

0 -      Sair
```

15. [Manzano, 1996] Escreva um programa em C para efetuar o cálculo da quantidade de litros de combustível gastos em uma viagem, utilizando-se um automóvel que faz 12 km por litro. Para obter o cálculo, o usuário deverá fornecer o tempo gasto na viagem e a velocidade média durante a mesma. Desta forma, será possível obter a distância percorrida com a fórmula $\text{distância} = \text{tempo} * \text{velocidade}$. Tendo o valor da distância, basta calcular a quantidade de litros de combustível utilizada da viagem com a fórmula $\text{litros_usados} = \text{distância} / 12$. O programa deverá apresentar os valores da velocidade média, tempo gasto na viagem, a distância percorrida e a quantidade de litros utilizada na viagem.

16. [Damas, 2007] Escreva um programa em C que apresente duas linhas com a *string* `Aqui vai um apito` ouvindo-se ao final de cada *string* em sinal sonoro.

17. [Mizrahi, 1990a] Escreva um programa em C que leia um caractere e apresente-o em decimal, octal e hexadecimal. Exemplo:

```
Digite um caractere: m
Caractere: m
Decimal: 109
Octal: 155
Hexadecimal: 6d
```

18. [Manzano, 1996] Escreva um programa em C para efetuar o cálculo do valor de uma prestação em atraso, utilizando a fórmula $\text{prestação} = \text{valor} + (\text{valor} * (\text{taxa} / 100) * \text{tempo})$.

19. [Mizrahi, 1990a] Escreva um programa em C que imprima o caractere fornecido pelo usuário, utilizando a função `getchar()`. Exemplo:

```
Digite um caractere: a
A tecla que você pressionou é a.
```

20. [Damas, 2007] Escreva um programa em C que indique qual o significado dos seguintes caracteres especiais: `\n`, `\\`, `\t`, `%%`.

21. [Manzano, 1996] Escreva um programa em C que leia dois inteiros (variáveis A e B) e efetue as operações de adição, subtração, multiplicação e divisão de A por B, apresentando ao final os quatro resultados obtidos.

22. [Manzano, 1996] Escreva um programa em C que leia quatro inteiros e apresente o resultado dois a dois da adição e da multiplicação, baseando-se na utilização da propriedade distributiva. Ou seja, se forem lidas as variáveis A, B, C e D, deverão ser somadas e multiplicadas A com B, A com C e A com D. Depois B com C, B com D, e por fim C com D e assim por diante.
23. [Damas, 2007] Escreva um programa em C que peça ao usuário dois inteiros e apresente o resultado da realização das operações aritméticas tradicionais.
24. [Manzano, 1996] Escreva um programa em C que leia dois inteiros para as variáveis A e B e efetue a troca dos valores, de forma que a variável A passe a possuir o valor da variável B e que a variável B passe a possuir o valor da variável A. Apresentar os valores trocados.
25. [Damas, 2007] Escreva um programa em C que solicite em determinado número de segundos e, em seguida, indique quantas horas, minutos e segundos esse valor representa.

26. [Mizrahi, 1990a] Considere o código

```
int x = 1, y = 2, z = 3;  
x += y += z += 7;
```

Quais serão os valores das variáveis x, y e z? Escreva um programa em C para checar a sua resposta.

27. [Damas, 2007] Escreva um programa em C que solicite um determinado número real e mostre qual a sua parte inteira e a sua parte fracionária.
28. [Manzano, 1996] Escreva um programa em C que leia um número inteiro e apresente o resultado do quadrado deste número.
29. [Manzano, 1996] Escreva um programa em C que leia dois inteiros (variáveis A e B) e imprima o resultado do quadrado da diferença do primeiro valor pelo segundo.
30. [Mizrahi, 1990a] Escreva um programa em C que converta a idade fornecida pelo usuário em anos para minutos. Exemplo:

```
Digite sua idade em anos: 12  
Sua idade em minutos e' 6307200.
```

31. [Salveti, 1998] Escreva um programa em C que leia um número inteiro e apresente o último dígito desse número. Considere que o número fornecido pelo usuário esteja na faixa de 0 a 9999. Por exemplo, se o usuário fornecer o número 7653, o programa deverá apresentar na saída o número 3, que é o último dígito do número fornecido pelo usuário.
32. [Salveti, 1998] Escreva um programa em C que dado um número inteiro de três algarismos, inverta a ordem de seus algarismos. Os três algarismos do número dado são diferentes de zero. Por exemplo, caso o usuário forneça o número 123 o programa deverá apresentar na saída o número 321.
33. [Damas, 2007] Escreva um programa que solicite ao usuário uma determinada data e a mostre em seguida no formato dd/mm/aaaa.

34. [Salveti, 1998] Escreva um programa em C que dado um número inteiro de três algarismos construa outro número de quatro algarismos de acordo com a seguinte regra: a) os três primeiros algarismos, contados da esquerda para a direita são iguais aos do número dado; b) o quarto algarismo é um dígito de controle calculado da seguinte forma: primeiro algarismo + segundo algarismo x 3 + terceiro algarismo x 5; o dígito de controle é igual ao resto da divisão dessa soma por 7. Por exemplo, se o número fornecido pelo usuário for 123, a saída deverá ser 1231.
35. [Mizrahi, 1990a] Assuma que todas as variáveis são do tipo `int`. Encontre o valor de cada uma delas e escreva um programa em C que as imprima para verificar os resultados.

```
x = (2 + 1) * 6;  
y = (5 + 1) / 2 * 3;  
i = j = (2 + 3) / 4;  
a = 3 + 2 * (b = 7 / 2);  
c = 5 + 10 % 4 / 2;
```

36. [Salveti, 1998] Escreva um programa em C que dado um número inteiro de segundos determine o seu valor equivalente em graus, minutos e segundos. Se a quantidade de segundos for insuficiente para dar o valor em graus, o valor em graus deve ser zero. A mesma observação continua válida em relação a minutos e segundos. Por exemplo: 3.600 segundos = 1 grau, 0 minutos, 0 segundos; 3.500 segundos = 0 graus, 58 minutos, 20 segundos.
37. [Mizrahi, 1990a] Qual será o valor de `k`? Escreva um programa em C que imprima o valor de `k`.

```
int j = 3;  
int k = j == 3;
```

38. [Salveti, 1998] Escreva um programa em C que dado duas frações `a/b` e `c/d`, determinar a sua soma e o seu produto.
39. [Salveti, 1998] Escreva um programa em C que dado um número inteiro representando um número binário de cinco dígitos, determinar o seu equivalente em decimal.
40. [Salveti, 1998] Escreva um programa em C que dado o primeiro termo e a razão de um progressão aritmética, determinar a soma dos seus primeiros cinco termos.

41. [Damas, 2007] Assinale a alternativa correta:

Os comentários devem ser escritos

- antes de qualquer instrução do programa.
- depois de todas as instruções.
- antes do `main`.
- sempre que o programador ache necessário ou conveniente.

42. [Mizrahi, 1990a] Escreva um programa em C que imprima na tela:

```
um  
dois  
três
```

43. [Damas, 2007] Assinale a alternativa correta:

Um programa em C, que tenha comentários no seu código, é, em relação a outro que não os tenha

- () mais rápido para executar.
- () mais lento para executar.
- () executado praticamente à mesma velocidade, pois os comentários exigem uma utilização ínfima da CPU.
- () executado à mesma velocidade, pois os comentários são simplesmente ignorados pelo compilador, não havendo qualquer reflexo deles no tempo de execução.

44. [Damas, 2007] A Indique se são verdadeiras ou falsas as seguintes afirmações.

Os comentários

- () só podem ocupar uma única linha.
- () podem ocupar várias linhas.
- () podem conter outros comentários dentro.
- () começam por /* e terminam com */.
- () não têm qualquer influência na velocidade de execução de um programa.
- () têm que começar no início de uma linha.
- () quando ocupa apenas uma linha não precisam terminar com */.

45. [Damas, 2007] Responda as seguintes questões:

- a) Qual a função que deve estar presente em todos os programas em C ?
- b) Como devem terminar todas as instruções em C?
- c) Como é delimitado um bloco em C?
- d) A função `printf` é parte integrante da linguagem C?
- e) Para que serve a linha `#include <stdio.h>` num programa?
- f) A extensão `.h` indica que o arquivo correspondente é composto por ...?
- g) Os arquivos com extensão `.h` são também conhecidos por ...?
- h) Dentro de uma *string* pode-se usar letras maiúsculas? Justifique.
- i) Qual o significado de `stdio`?

46. [Damas, 2007] Identifique os erros de compilação que seriam detectados nos seguintes programas:

```
/*
 * Programa a)
 */
#include <stdio.h>
Main()
{
    printf("Hello World");
}

/*
 * Programa b)
 */
#include <stdio.h>
main
{
    printf("Hello World");
}

/*
 * Programa c)
 */
#include <stdio.h>
main()
{
    print ("Hello World");
}

/*
 * Programa d)
 */
#include <stdio.h>
main()
{
    printf("Hello") ("World");
}

/*
 * Programa e)
 */
#include <stdio.h>
main()
{
    printf("Hello World");
}

/*
 * Programa f) */
 */
#include <stdio.h>
main()
{
    printf("Hello World");
}
```

```
/*
 * Programa g)
 */
#include <stdio.h>
main()
{
    printf>Hello World);
}

/*
 * Programa h)
 */
#include <stdio.h>
main()
{
    printf("Hello World")
}

/*
 * Programa i)
 */
include <stdio.h>
main()
{
    printf("Hello World");
}

/*
 * Programa j)
 */
#include <stdio.h>
main()
{
    printf('Hello World');
}
```

47. [Mizrahi, 1990a] Qual será a saída dos programas abaixo:

```
/*
 * Programa a)
 */
#include <stdio.h>
main()
{
    printf("Este e' o numero dois: %d", 2);
}

/*
 * Programa b)
 */
#include <stdio.h>
main()
{
    printf("%s esta a %d milhões de milhas\ndo sol", "Vênus", 67);
}
```

```
/*
 * Programa c)
 */
#include <stdio.h>
main()
{
    printf("A letra %c ", 'j');
    printf("pronuncia-se %s.", "jota");
}

/*
 * Programa d)
 */
#include <stdio.h>
main()
{
    int evento;
    char corrida;
    float tempo;
    evento = 5;
    corrida = 'C';
    tempo = 27.25;
    printf("O tempo vitorioso na eliminatória %c", corrida);
    printf("\nda competição %d foi %f.", evento, tempo);
}

/*
 * Programa e)
 */
#include <stdio.h>
main()
{
    int reajuste = 10;
    printf("O reajuste foi de %d%%.", reajuste);
}
```

48. [Mizrahi, 1990a] Qual será a saída dos programas abaixo:

```
/*
 * Programa a)
 */
#include <stdio.h>
main()
{
    printf("Os alunos são %2d.\n", 350);
    printf("Os alunos são %4d.\n", 350);
    printf("Os alunos são %5d.\n", 350);
}

/*
 * Programa b)
 */
#include <stdio.h>
main()
{
    printf("%4.2f\n", 3456.78);
    printf("%3.2f\n", 3456.78);
    printf("%3.1f\n", 3456.78);
    printf("%10.3f\n", 3456.78);
}
```



```
/*
 * Programa c)
 */
#include <stdio.h>
main()
{
    printf("%.2f %.2f %.2f\n", 8.0, 15.3, 584.13);
    printf("%.2f %.2f %.2f\n", 834.0, 1500.55, 4890.21);
}

/*
 * Programa d)
 */
#include <stdio.h>
main()
{
    printf("%10.2f %10.2f %10.2f\n", 8.0, 15.3, 584.13);
    printf("%10.2f %10.2f %10.2f\n", 834.0, 1500.55, 4890.21);
}

/*
 * Programa e)
 */
#include <stdio.h>
main()
{
    printf("\n%04d", 21);
    printf("\n%06d", 21);
    printf("\n%6.4d", 21);
    printf("\n%6.0d", 21);
}

/*
 * Programa f)
 */
#include <stdio.h>
main()
{
    printf("%d %c %x %o \n", 'A', 'A', 'A', 'A');
    printf("%c %c %c %c \n", 'A', 65, 0x41, 0101);
}
```

49. [Mizrahi, 1990a] Qual será a saída dos programas abaixo:

```
/*
 * Programa a)
 */
#include <stdio.h>
main() {
    int num = 0;
    printf("%d e' um belo numero\n", num);
    printf("%d e' um belo numero\n", num++);
    printf("%d e' um belo numero\n", num);
}
```

```
/*
 * Programa b)
 */
#include <stdio.h>
main() {
    int num = 0;
    printf("%d e' um belo numero\n", num);
    printf("%d e' um belo numero\n", ++num);
    printf("%d e' um belo numero\n", num);
}

/*
 * Programa c)
 */
#include <stdio.h>
main() {
    int num = 0;
    printf("%d e' um belo numero\n", num);
    printf("%d e' um belo numero\n", num--);
    printf("%d e' um belo numero\n", num);
}

/*
 * Programa d)
 */
#include <stdio.h>
main() {
    int num = 0;
    printf("%d e' um belo numero\n", num);
    printf("%d e' um belo numero\n", --num);
    printf("%d e' um belo numero\n", num);
}
```

50. [Damas, 2007] Indique quais das seguintes declarações estão corretas.

- () `y int;`
- () `int;`
- () `integer x;`
- () `inta, b;`
- () `float f, g, c;`
- () `char ch1=ch2='A';`
- () `char ch1 = 'A', ch2 = 'A';`

51. [Damas, 2007] Uma variável inteira, quando é declarada, é sempre iniciada com?

- () 0 (zero)
- () 1 (um)
- () um valor aleatório
- () um valor negativo

52. [Damas, 2007] Indique, na seguinte lista, quais os identificadores corretos e incorretos de variáveis.

- () Valor
- () &xvar
- () dez%
- () a+b
- () _Kabonga
- () MENOS
- () 10a
- () a10
- () main
- () F1

53. [Damas, 2007] O nome de uma variável...

- () deve indicar aquilo que ela armazena
- () deve ser o menor possível
- () deve ser o maior possível
- () deve ser o mais explícito possível
- () deve ser todo escrito em maiúsculas
- () pode, mas não deve, começar com underscore (`_`)
- () pode conter mais do que um caractere

54. [Damas, 2007] Associe os seguintes tipos aos correspondentes formatos de leitura e escrita.

| | |
|------------------------------|------------------|
| <code>int</code> | <code>%e</code> |
| <code>float</code> | <code>%ld</code> |
| <code>char</code> | <code>%f</code> |
| <code>short int</code> | <code>%d</code> |
| <code>long int</code> | <code>%hd</code> |
| <code>signed long int</code> | <code>%c</code> |

55. [Damas, 2007] Indique quais das seguintes afirmações são verdadeiras e quais são falsas.

- () O tipo `float` reserva espaço em memória para um real com precisão simples, enquanto o tipo `double` reserva espaço para uma variável com precisão dupla.
- () O tipo `char` pode ter os prefixos `long` e `short`.
- () O tipo `char` pode ter os prefixos `signed` e `unsigned`.
- () Uma variável declarada como `unsigned` pode comportar valores superiores a uma outra que seja declarada como `signed`.

- () Uma variável do tipo `char` pode armazenar caracteres individuais ou conjuntos de caracteres também denominados por *strings*.
- () Uma variável do tipo `char` pode armazenar vários caracteres, desde que todos eles sejam caracteres ASCII.
- () Uma variável do tipo `char` pode armazenar vários caracteres, desde que sejam caracteres especiais.
- () O operador módulo (%) não pode ser utilizado em reais.

56. [Damas, 2007] Identifique os erros de compilação que seriam detectados nos seguintes programas.

```
/*
 * Programa a)
 */
#include <stdio.h>
main()
{
    int x, y, x;
}

/*
 * Programa b)
 */
#include <stdio.h>
main()
{
    int x, y;
    float int = 5.23;
    printf("%f", int);
}

/*
 * Programa c)
 */
#include <stdio.h>
main()
{
    int x = y = z = 0;
    printf("%d %d %d \n", x, y, z);
}
```

57. [Damas, 2007] Embora os programas que se seguem não tenham erros de compilação, identifique as causas de seu possível mau funcionamento.

```
/*
 * Programa a)
 */
#include <stdio.h>
main() {
    int n;
    scanf("Introduza um numero: %d", &n);
    printf("O numero eh %d\n", n);
}
```

```
/*
 * Programa b)
 */
#include <stdio.h>
main() {
    int n;
    printf("Introduza um numero: ");
    scanf("%d\n", &n);
    printf("O numero eh %d\n", n);
}

/*
 * Programa c)
 */
#include <stdio.h>
main() {
    int n;
    printf("Introduza um numero: ");
    scanf("%f", &n);
    printf("O numero eh %f\n", n);
}
```