

Lista 1 (EXERCÍCIOS)

1) Converta os seguintes números na base apresentada para as bases 2, 8, 10 e 16.

a) 11011001 (base 2)

b) 777 (base 8)

c) FEC (base 16)

d) 27 (base 10)

2) Explique como as bases 8 e 16 são relacionadas com a base 2.

3) Considere o programa abaixo:

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

```
int main()
```

```
{
```

```
    char str[5] = {0};
```

```
    char valor = 10;
```

```
    scanf("%s", &str);
```

```
    printf("String: %s\nValor:%d\n", str, valor);
```

```
    return 0;
```

```
}
```

a) Qual a saída para uma entrada "test" ?

b) Qual a saída para uma entrada "teste" ?

Explique.

4) Qual a saída do seguinte programa?

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

```
int main()
```

```
{
```

```
    printf("\nab");
```

```
    printf("\bsi");
```

```
    printf("\rha");
```

```
    return 0;
```

```
}
```

5) Qual o valor de “char a = 1200”? Qual o nome do erro que leva a esse comportamento?

6) Monte o teste de mesa e responda qual o valor final de x, y e z após todas operações.

```
int x,y;
```

```
int a = 14, b = 3;
```

```
float z;
```

```
x = a/b;
```

```
y = a%b;
```

```
z = y/x;
```

7) Faça um programa que leia um valor inteiro de até 4 algarismos e calcule a soma dos algarismos desse número. Caso o valor informado seja maior que 4 algarismos, emita um erro. Exemplo:

Informe um número:

3267

Soma dos algarismos do número:

18

8) Palíndromos são números que têm o mesmo valor se lidos da esquerda para a direita ou da direita para a esquerda. Exemplos: 44, 232, etc. Faça um programa que determine e escreva todos os números inteiros entre 1000 que 10000 que são palíndromos.

9) Explique a saída do seguinte programa:

```
#include <stdio.h>

int main()
{
    {
        int a[100];
        printf("%X\n", &a);
    }
    {
        char b[100*sizeof(int)];
        printf("%X\n", &b);
    }
    return 0;
}
```

10) Descreva o que acontece em todas as linhas do seguinte programa e qual é a saída:

```
#include <stdio.h>

int main()
{
    char a[5][5] = {0};
    printf("%d %d %d\n", sizeof(a), sizeof(a[0]), sizeof(a[0][0]));
    printf("%X %X %d\n", a, a[0], a[0][0]);
    printf("%X %X %d\n", a, a[1], a[1][0]);
    printf("%X %X %X %X %X\n", a[0], a[1], a[2], a[3], a[4]);
    return 0;
}
```

11) Considere os dois vetores a seguir:

```
char str[][3] = {"aa","bb","cc","dd"};
```

```
char *str[] = {"aa","bb","cc","dd"};
```

Eles são iguais? Crie uma função que receba o vetor como parâmetro e imprima as quatro strings. Isso consegue ser realizado com apenas um protótipo ou são necessários protótipos diferentes para cada sintaxe de vetor? Explique.

12) Crie uma função `novo_inteiro()` que declara e inicializa um inteiro **DENTRO DA FUNÇÃO** e retorna o endereço para esse inteiro. Imprima o valor do inteiro e o endereço. Isso é um código válido? O compilador exibe avisos? Essa é uma operação segura?

13) Escreva uma função que receba quatro parâmetros, o tamanho de passo, o intervalo mínimo, o intervalo máximo e o ponteiro para uma função que será integrada. Exemplo: a integral de x^2 no intervalo de (0,5) é 41,67.

Dica a integral contínua: $\int_a^b f(x)dx$ pode ser aproximada pelo somatório $\sum_a^b f(x)dx$.

14) Escreva um programa que leia um arquivo contendo sequências de tuplas (nome, número), separadas por linhas. Considere um máximo de 256 entradas, cada nome com um máximo de 255 caracteres.

15) Modifique o programa anterior. Desta vez, inclua uma rotina que ordene os registros de acordo com os valores numéricos, em ordem decrescente. Os registros ordenados deverão ser salvos substituindo o arquivo original.

Exemplo:

João 14

Pedro 3

José 43

Saída:

Pedro 3

João 14

José 43

Dica: você pode escrever sua própria rotina de ordenação, entretanto, irá agregar mais conhecimento se aprender a utilizar a rotina `qsort`.

16) Utilize conversão de tipos e conserte o código abaixo para que ele resulte na saída esperada:

```
int x = 9, y = 10;
float z = x/y;
printf("%f\n", z);
```

17) O código abaixo não funciona como o esperado. Identifique e corrija o erro. Porque isso ocorre?

```
int i = -10;
unsigned int stop_val = 0;

while(i <= stop_val)
{
    printf("%d\n", i);
    i = i + 1;
}
```

18) Crie uma biblioteca consistindo de ponto.h e ponto.c Defina uma estrutura Ponto que represente pontos no plano Cartesiano. Escreva funções para medir a distância entre dois pontos e comparar se dois pontos são iguais.