

Lista 3 (EXERCÍCIOS)

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

- 2) Qual o valor decimal de $B0_{16}$, assumindo um valor de 8 bits, considerando o bit de sinal e a representação por complemento de dois.

- 3) Um valor de 16 bits é dado por (01111111 11111111), responda:
 - a) Qual o valor decimal.
 - b) Qual o valor decimal se o primeiro bit for “1” e considerando a representação por magnitude assinalada.
 - b) Qual o valor decimal se o primeiro bit for “1” e considerando a representação por complemento de dois.

- 4) Escreva os seguintes inteiros em big endian:
 - a) 0xFACEB00C
 - b) 0xDEADBEEF

- 5) Escreva os seguintes inteiros em little endian:
 - a) 0xCEFAEDFE
 - b) 0xE011CFD0
 - c) 0x5EEA150D
 - d) 0xDEC0ADDE

- 6) Considerando alguma determinada arquitetura, sistema operacional e compilador, qual o valor de sizeof para:
 - a) char
 - b) short int
 - b) int
 - c) long int
 - d) float
 - e) double
 - f) long double
 - g) void*

7) Qual o valor de 0.15625 em binário?

8) Qual o valor de 4.625 em binário?

9) Qual o valor de 0.1 em binário?

10) Por qual motivo alguns valores decimais não podem ser representados em binário sem aproximações?

11) Qual a função de void como parâmetro de uma função ou retorno?

12) Para que serve e o que significa um ponteiro do tipo void?

13) Compile um programa simples usando o GCC e a flag `-S` com apenas duas declarações no *main* “`char a = 120; printf(“%X\n”,a);`”. Esse programa irá gerar um *assembly* similar a:

```
movb $120, 31(%esp)
```

```
movsbl 31(%esp), %eax
```

```
movl %eax, 4(%esp)
```

```
movl $LC0, (%esp)
```

```
call _printf
```

Explique a relação entre a instrução `movsbl` e a promoção para inteiro.

14) Modifique o programa anterior, dessa vez defina a variável “a” como *register char*. O que muda no *assembly*?

15) Escreva um programa que imprima na tela os seguintes valores em decimal, fazendo uso de literais na chamada de `printf`:

a) 49374 (hexadecimal)

b) 123456 (octal)

16) Escreva um programa que receba um valor de entrada e imprima “1” caso o valor seja par ou “0” para ímpar. Use apenas operadores aritméticos, lógicos ou binários, **não use IF**.

17) Escreva um programa que receba duas entradas e imprima “1” caso um valor seja divisível pelo outro e “0” caso contrário. Use apenas operadores aritméticos, lógicos ou binários, **não use IF**.

18) Qual das declarações abaixo está errada?

- a) int i;
- b) long float x;
- c) long double y;
- d) long ijk;
- e) short int a;
- f) unsigned b;

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

```
int x,y,z;  
x=y=10;  
z=++x;  
x=-x;  
y++;  
x=x+y-(z--);
```

20) 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;
```

21) Qual o resultado da seguinte operação lógica?

$(-5 \parallel 0) \&\&(3 \geq 2) \&\&(1 \neq 0) \parallel (3 < 0)$

22) Quais os valores de a, b e c após a execução do seguinte código?

```
int a = 10, b = 20, c;
```

```
c = a+++b;
```

23) Altere a segunda linha do exercício anterior para “c = a+++++b;”, o programa compila? As alternativas “c = a++ +++b;” ou “c = a+++ ++b;” funcionam? Explique.

(dica: pesquise por maximal munch rule)

24) Escreva um programa que receba três entradas, um valor, uma posição de bit e uma operação. O programa deve executar a operação escolhida sobre o valor:

a) Escrever um bit a posição indicada.

b) Apagar um bit na posição indicada.

c) Ler um bit na posição indicada.

25) Desenvolva um programa que leia um valor positivo e escreva o resultado do complemento de dois sobre esse valor. Apresente a saída em decimal e hexadecimal.

26) Desenvolva um programa que leia um valor negativo e escreva o resultado do inverso de complemento de dois sobre esse valor. Apresente a saída em decimal e hexadecimal.

27) Escreva um programa que receba um número e imprima “1” se o número for negativo ou zero, imprime “0” caso contrário (positivo). Use apenas operadores aritméticos, lógicos ou binários, **não use IF**.

28) Escreva um programa que receba dois números e imprima “1” caso tenham sinais opostos e “0” caso tenham o mesmo sinal. Use apenas operadores aritméticos, lógicos ou binários, **não use IF**.

29) Escreva um programa que receba um valor inteiro e imprima seu resultado em binário, utilizando apenas operadores binários.

30) Descreva um algoritmo utilizando XOR (^) para criptografar e descriptar uma string.