

Roteiro 2 - Laboratório de Programação 2

1.1 -

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

typedef struct{
    int numero;
    float saldo;
    char titular[20];
} ContaBancaria;

void criarConta(ContaBancaria *c, int numero, char *titular){
    c->numero = numero;
    c->saldo = 0;
    strcpy(c->titular, titular);
}

void depositar (ContaBancaria *c, double deposito){
    if (deposito < 0){
        printf("Deposito invalido\n");
    }
    if (deposito == 0){
        return;
    }
    c->saldo += deposito;
}

void sacar (ContaBancaria *c, double saque){
    if (c->saldo < saque){
        printf("Saque invalido\n");
    } else {
        c->saldo -= saque;
    }
}

double consultarSaldo(ContaBancaria *c){
    return c->saldo;
}

void imprimirInfo(ContaBancaria *c){
```

```

        printf("Titular: %s\n", c->titular);
        printf("Numero: %d\n", c->numero);
        printf("Saldo: %.2f\n", c->saldo);
    }

int main(){
    ContaBancaria c;
    criarConta(&c, 123, "Gabriel");
    depositar(&c, 100);
    imprimirInfo(&c);
    sacar(&c, 50);
    imprimirInfo(&c);

    return 0;
}

```

Saída:

```

Titular: Gabriel
Numero: 123
Saldo: 100.00
Titular: Gabriel
Numero: 123
Saldo: 50.00

```

1.2 -

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct{
    char nome[20];
    int quantidade;
    float preco;
} Produto;

typedef struct{
    Produto produtos[100];
    int totalProdutos;
} CatalogoProdutos;

void criarCatalogo(CatalogoProdutos *c){
    c->totalProdutos = 0;
}

```

```

void adicionarProduto(CatalogoProdutos *c, char *nome, double preco,
int quantidade){
    strcpy(c->produtos[c->totalProdutos].nome, nome);
    c->produtos[c->totalProdutos].preco = preco;
    c->produtos[c->totalProdutos].quantidade = quantidade;
    c->totalProdutos++;
}

int verificarEstoque(CatalogoProdutos *c, char *nome){
    for (int i = 0; i < c->totalProdutos; i++){
        if (strcmp(c->produtos[i].nome, nome) == 0){
            printf("Estoque de %s: %d\n", c->produtos[i].nome,
c->produtos[i].quantidade);
            return c->produtos[i].quantidade;
        }
    }

    return -1;
}

void imprimirCatalogo(CatalogoProdutos *c){
    for (int i = 0; i < c->totalProdutos; i++){
        printf("%s - %.2f - %d\n", c->produtos[i].nome,
c->produtos[i].preco, c->produtos[i].quantidade);
    }
}

int main(){
    CatalogoProdutos catalogo;
    criarCatalogo(&catalogo);

    adicionarProduto(&catalogo, "Arroz", 2.50, 10);
    adicionarProduto(&catalogo, "Feijao", 3.50, 5);
    imprimirCatalogo(&catalogo);
    verificarEstoque(&catalogo, "Arroz");
    verificarEstoque(&catalogo, "Feijao");

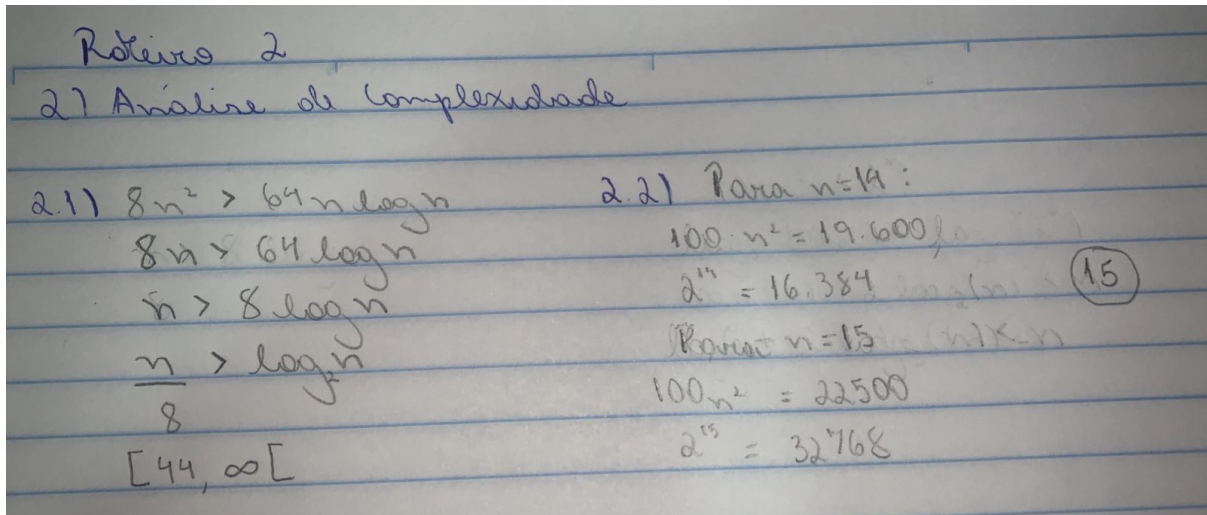
    return 0;
}

```

Saída:

```
Arroz - 2.50 - 10
Feijao - 3.50 - 5
Estoque de Arroz: 10
Estoque de Feijao: 5
```

2.1 / 2.2 -



2.3 - Quando uma função $g(n)$ é $O(f(n))$, a função $f(n)$ impõe um limite superior assintótico para $g(n)$. Isso é, multiplicado por uma constante c , a função $f(n)$ limita superiormente a função $g(n)$, a partir de um determinado n_0 .

2.4 - Quando uma função $g(n)$ é $\Omega(f(n))$, a função $f(n)$ impõe um limite inferior assintótico para $g(n)$. Isso é, multiplicado por uma constante c , a função $g(n)$ nunca será inferior a função $f(n)$, a partir de um determinado n_0 .

2.5 - A declaração não faz sentido, porque utilizar a palavra "mínimo" não é correto, por se tratar de um limite superior assintótico.

2.6 / 2.7 / 2.8 -

$$2.6) a(n) = n^2 - n + 500$$

$$b(n) = 47n + 47$$

$$n^2 - n + 500 < 47n + 47$$

$$n^2 - 48n + 453 < 0$$

$$\Delta = \sqrt{48^2 - 4 \cdot 453}$$

$$\Delta \approx 22,2$$

$$n = \frac{48 \pm 22,2}{2}$$

$$2$$

$$n = \frac{48 \pm 22,2}{2}$$

$$12,9 < n < 35,1$$

$$[13, 35]$$

2.7) Loop mais interno: $3 \cdot 1$, indo de 1 a 3

Loop do meio: $N-1$, indo de 1 a N

Loop mais externo: $N-1$, indo de 0 a $N-1$

$$(N-1) \cdot (N-1) \cdot (N-2)$$

$$= O(N^3)$$

2.8) Para descobrir o maior elemento

de um vetor, mesmo que o maior elemento seja o primeiro, o percorremos por inteiro ($N-1$ passos).

Dadas duas constantes c e m , temos:
para valores $n \gg m \rightarrow \text{maior} \leq c \cdot f(m)$.

Maior é $O(n)$

Ainda, se existirem c e m tais que

$\text{maior} \geq c \cdot f(m) \quad \forall n \gg m$.

Maior é $\Omega(n)$

Se é $O(n)$ e $\Omega(n)$ é $\Theta(n)$