Lista de Laboratório de Programação II

Nome : Messias Feres Curi Melo

Matrícula : 2022003764

Questão 1.1

Código em C:

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

typedef struct{
    int numeroConta;
    double saldo;
    char* nomeTitular;
}ContaBancaria;

void criarConta(ContaBancaria* c, int numero, char *titular){
    c->numeroConta = numero;
    c->saldo = 0.0;
    c->nomeTitular = titular;
}

void depositar(ContaBancaria *c, double valor){
    c->saldo += valor;
}

void sacar(ContaBancaria *c, double valor){
    c->saldo -= valor;
}

double consultarSaldo(ContaBancaria *c){
    printf("Seu saldo é : %.2lf reais\n", c->saldo);
}

void imprimirInfo(ContaBancaria *c){
    printf("Imprimindo Informações\nNúmero da Conta : %d\nNome do Titular : %s\nSaldo : %.2lf reais\n ", c->numeroConta, c->nomeTitular, c->saldo);
}


int main(){
    ContaBancaria *c = (ContaBancaria*)malloc(sizeof(ContaBancaria));
    criarConta(c,10,"Messias");
    depositar(c, 149.99);
    sacar(c, 49.99);
    consultarSaldo(c);
    imprimirInfo(c);
    free(c);

    return 0;
}
```

Exibição no terminal:

```
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho$ gcc tp1.1.c -o tp11
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho$ ./tp11
Seu saldo é : 100.00 reais
Imprimindo Informações
Número da Conta : 10
Nome do Titular : Messias
Saldo : 100.00 reais
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho$
```

Questão 1.2

Código em C

C: > Users > Messi > OneDrive > Área de Trabalho > C tp1.2.c > criarCatalogo(CatalogoProduto *)

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

typedef struct Produto{
    char nome[30];
    double preco;
    int quantidade;
}Produto;

typedef struct CatalogoProduto{
    Produto produtos[100];
    int numeroDeProdutos;
}CatalogoProduto;

void criarCatalogo(CatalogoProduto *c){
    c->numeroDeProdutos = 0;
    printf("Criado com sucesso!\n");
}

void adicionarProduto(CatalogoProduto *c, char *nome, double preco, int quantidade){
    if (c->numeroDeProdutos < 100) {
        strcpy(c->produtos[c->numeroDeProdutos].nome, nome);
        c->produtos[c->numeroDeProdutos].preco = preco;
        c->produtos[c->numeroDeProdutos].quantidade = quantidade;
        c->numeroDeProdutos++;
    }else{
        printf("Máximo de produtos alcançado\n");
    }
}

int verificarEstoque(CatalogoProduto *c, char *nome) {
    for (int i = 0; i < c->numeroDeProdutos; i++) {
        if (strcmp(c->produtos[i].nome, nome) == 0) {
            return c->produtos[i].quantidade;
        }
    }
    return 0;
}

void imprimirCatalogo(CatalogoProduto *c) {
    for (int i = 0; i < c->numeroDeProdutos; i++) {
        printf("Produto %d\n", i + 1);
        printf("Nome : %s\n", c->produtos[i].nome);
        printf("Preço : %.2lf reais\n", c->produtos[i].preco);
        printf("Quantidade : %d\n", c->produtos[i].quantidade);
```

```c
    }
}

int main(){
    int escolha, quantidade;
    char nome[30];
    double preco;
    CatalogoProduto *c = (CatalogoProduto *)malloc(sizeof(CatalogoProduto));
    while (true){
        printf("1 - Criar Catálogo\n");
        printf("2 - Adicionar Produto\n");
        printf("3 - Verificar Estoque\n");
        printf("4 - Imprimir Catálogo\n");
        printf("0 - Sair\n");
        printf("Escolha : ");
        scanf("%d", &escolha);

        if (escolha == 1){
            criarCatalogo(c);
        }else if (escolha == 2){
            printf("Nome : ");
            scanf("%s", nome);
            printf("Preço : ");
            scanf("%lf", &preco);
            printf("Quantidade : ");
            scanf("%d", &quantidade);
            adicionarProduto(c, nome, preco, quantidade);
        }else if (escolha == 3){
            printf("Nome : ");
            scanf("%s", nome);
            int quant = verificarEstoque(c, nome);
            printf("Quantidade em estoque : %d\n", quant);
        }else if (escolha == 4){
            imprimirCatalogo(c);
        }else if (escolha == 0){
            break;
        }else{
            printf("Escolha inválida\n");
        }
    }
    free(c);
    return 0;
}
```

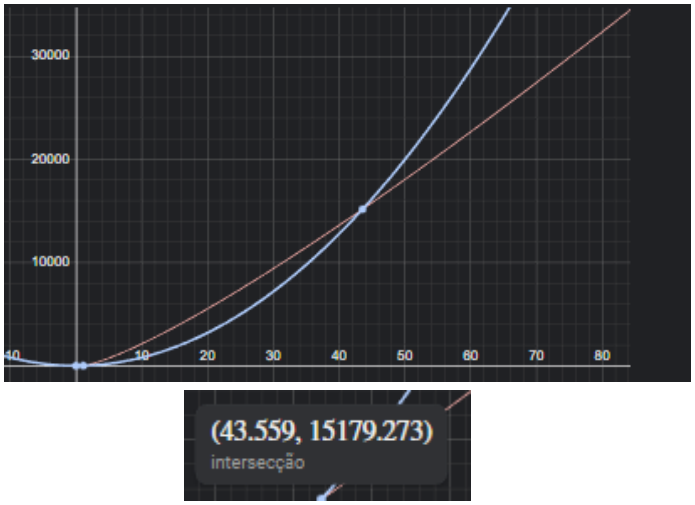
Exibição no terminal:

```
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho$ ./tp12
1 - Criar Catálogo
2 - Adicionar Produto
3 - Verificar Estoque
4 - Imprimir Catálogo
0 - Sair
Escolha : 1
Criado com sucesso!
1 - Criar Catálogo
2 - Adicionar Produto
3 - Verificar Estoque
4 - Imprimir Catálogo
0 - Sair
Escolha : 2
Nome : Bola
Preço : 23.59
Quantidade : 12
1 - Criar Catálogo
2 - Adicionar Produto
3 - Verificar Estoque
4 - Imprimir Catálogo
0 - Sair
Escolha : 3
Nome : Bola
Quantidade em estoque : 12
1 - Criar Catálogo
2 - Adicionar Produto
3 - Verificar Estoque
4 - Imprimir Catálogo
0 - Sair
Escolha : 4
Produto 1
Nome : Bola
Preço : 23.59 reais
Quantidade : 12
1 - Criar Catálogo
2 - Adicionar Produto
3 - Verificar Estoque
4 - Imprimir Catálogo
0 - Sair
Escolha : 0
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho$
```

Questão 2.1

$8(n^2) <= 64n \lg(n)$
$(n^2) <= 8n \lg(n)$
$n <= 8 \lg(n)$
$n/(\lg(n)) <= 8$
$n <= 43$

Tabela e gráfico abaixo para demonstração.

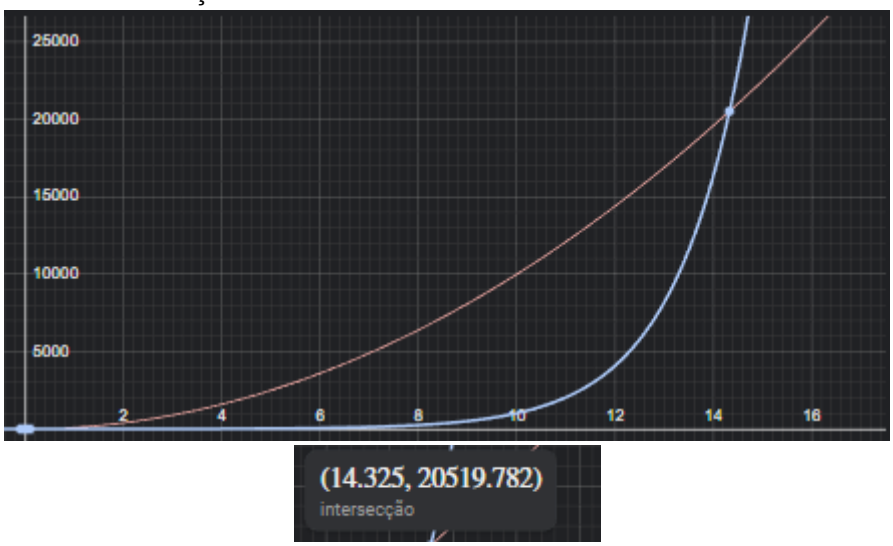


Valor de n	Inserção ($8n^2$)	Intercalação ($64n \log n$)
1	8	-
2	32	128
3	72	304,31
4	128	512
5	200	743,02
6	288	992,63
7	392	1257,70
8	512	1536
	...	
39	12168	13192,36
40	12800	13624,14
41	13448	14058,22
42	14112	14494,55
43	14792	14933,08

Questão 2.2

(100n^2) <= (2^n)
lg(100n^2) <= n
lg(100) + 2lgn <= n
6.65 + 2lgn <= n
n <= 14.325

Resposta: quando **n > 14.325**, o polinomial será melhor que o exponencial, gráfico abaixo para demonstração

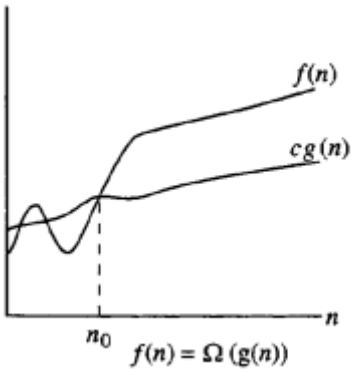


Questão 2.3

Formalmente, a função **O(f(n))** é um **limitante assintótico superior** para g(n).
Para todos os valores de n à direita de n_0, o valor de g(n) reside em c·f(n) ou abaixo desse.

Questão 2.4

g(n) é **Ω(f(n))** significa que **g(n)** é assintoticamente **limitada inferiormente** por uma função f(n) multiplicada por uma constante positiva, para valores suficientemente grandes de n.
Para todos os valores de n à direita de n_0, o valor de g(n) reside em c·f(n) ou acima desse.



(Imagem com o f invertido com o g)

Questão 2.5

Pois a notação O é usada para calcular um **limite superior** ou uma cota superior assintótica em relação ao crescimento do tempo de execução de um algoritmo à medida que a entrada (n) aumenta e **não medir um mínimo** ou no caso um limite inferior.

Questão 2.6

n^2-n+500 < 47n+47
n^2-48n+453 < 0
delta = 2304 - 1812 = 492(raiz)
n1 = (48+22)/2 => **35**
n2 = (48-22)/2 => **13**

Então o algoritmo A leva menos tempo que B quando 'n' está próximo do intervalo de **13 < n < 35**, sendo o A mais eficiente nesse intervalo.

Questão 2.7

A complexidade será O(n^3), por ter 3 for aninhados essa será a complexidade do pior caso.

```
{
int i,j,k,s;
for(i=0; i < N-1; i++)
  for(j=i+1; j < N; j++)
    for(k=1; k < j; k++)
      s = 1;
}
```

Questão 2.8

```
1  int maior(int*v, int n){
2      int i, MAX;
3      MAX = v[0];
4      for(i=1; i<n; i++)
5          if(v[i] > MAX)
6              MAX = v[i];
7      return MAX;
8  }
```

(v[i] > MAX)

O(n) - o algoritmo é **O(n)**, já que seu pior caso, ele cresce linearmente com o tamanho da entrada.

Ω(n) - o algoritmo é **Ω(n)**, já que seu melhor caso, também cresce linearmente com o tamanho da entrada.

Θ(n) - o algoritmo possui uma complexidade de tempo linear em todos os casos. Como já mostrado, o algoritmo é O(n) e Ω(n), o que significa que ele é **Θ(n)**.