

Professor Pietro Martins de Oliveira

AL GO RIT MOS

do início ao fim

***PACOTE DE EXERCÍCIOS 1:
ESTRUTURA SEQUENCIAL EM C***

- 1) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba quatro números inteiros, calcule e mostre a soma desses números.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída - `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada - `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 1 – Caso de teste

Informe quatro números, em sequência:

1
2
3
4

Resultado da soma: 10

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURA SEQUENCIAL – Exercício 1 – Solução

```
01 #include <stdio.h>
02 int main(){
03     int n1, n2, n3, n4, s;
04     printf("Informe quatro números, em sequência:\n");
05     scanf("%d", &n1);
06     scanf("%d", &n2);
07     scanf("%d", &n3);
08     scanf("%d", &n4);
09     s = n1 + n2 + n3 + n4;
10     printf("Resultado da soma: %d.\n", s);
11 }
```

- 2) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba três notas, calcule e mostre a média aritmética entre elas.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída - `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada - `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 2 – Caso de teste

Informe três notas, em sequência:

5,5
6,0
6,5

Média aritmética: 6

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURA SEQUENCIAL – Exercício 2 – Solução

```
01 #include <stdio.h>
02 int main(){
03     float n1, n2, n3, m;
04     printf("Informe três notas, em sequência:\n");
05     scanf("%f", &n1);
06     scanf("%f", &n2);
07     scanf("%f", &n3);
08     m = (n1+n2+n3)/3;
09     printf("Média aritmética: %f.\n", m);
10 }
```

- 3) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba três notas e seus respectivos pesos, calcule e mostre a média ponderada dessas notas.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída - printf ou puts). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada - scanf, gets ou fgets).

Exemplo de execução – Exercício 3 – Caso de teste

```
Insira a primeira nota:
10
Insira o peso da primeira nota:
2
Insira a segunda nota:
10
Insira o peso da segunda nota:
3
Insira a terceira nota:
10
Insira o peso da terceira nota:
5
Média ponderada: 10
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURA SEQUENCIAL – Exercício 3 – Solução

```
01 #include <stdio.h>
02 int main(){
03     float n1, n2, n3, p1, p2, p3, m;
04     printf("Insira a primeira nota:\n");
05     scanf("%f", &n1);
06     printf("Insira o peso da primeira nota:\n");
07     scanf("%f", &p1);
08     printf("Insira a segunda nota:\n");
09     scanf("%f", &n2);
10     printf("Insira o peso da segunda nota:\n");
11     scanf("%f", &p2);
12     printf("Insira a terceira nota:\n");
13     scanf("%f", &n3);
14     printf("Insira o peso da terceira nota:\n");
15     scanf("%f", &p3);
16     m = (n1*p1 + n2*p2 + n3*p3) / (p1+p2+p3);
17     printf("Média ponderada: %f.\n", m);
18 }
```

- 4) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba o salário de um funcionário, calcule e mostre o novo salário, sabendo-se que este sofreu um aumento de 25%.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída - printf ou puts). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada - scanf, gets ou fgets).

Exemplo de execução - Exercício 4 - Caso de teste

Insira o salário:

1000

Novo salário: 1250

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURA SEQUENCIAL - Exercício 4 - Solução

```
01 #include <stdio.h>
02 int main(){
03     float sal, novo_sal;
04     printf("Insira o salário:\n");
05     scanf("%f", &sal);
06     novo_sal = sal + sal*(25.0/100.0);
07     printf("Novo salário: %.2f.\n", novo_sal);
08 }
09 }
```

- 5) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba o salário de um funcionário e o percentual de aumento, calcule e mostre o valor do aumento e o novo salário.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída - `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada - `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 5 - Caso de teste

```
Insira o salário:
1000
Insira o percentual de aumento:
10
Valor do aumento: 100
Novo salário: 1100
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURA SEQUENCIAL – Exercício 5 – Solução

```
01 #include <stdio.h>
02 int main(){
03     float sal, perc, aumento, novo_sal;
04     printf("Insira o salário:\n");
05     scanf("%f", &sal);
06     printf("Insira o percentual de aumento:\n");
07     scanf("%f", &perc);
08     aumento = sal * perc/100.0;
09     novo_sal = sal + aumento;
10     printf("Valor do aumento: %.2f.\n", aumento);
11     printf("Novo salário: %.2f.\n", novo_sal);
12 }
```

- 6) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba o salário-base de um funcionário, calcule e mostre o salário a receber, sabendo-se que esse funcionário tem gratificação de 5% sobre o salário-base e paga imposto de 7% sobre o salário-base.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída - `printf` ou `puts`). Já os textos que estão em

branco correspondem a dados informados pelo usuário (operações de entrada - `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 6 – Caso de teste

Insira o salário base:

1000

Salário a receber: 980

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURA SEQUENCIAL – Exercício 6 – Solução

```
01 #include <stdio.h>
02 int main() {
03     float sal, liquido;
04     printf("Insira o salário base:\n");
05     scanf("%f", &sal);
06     liquido = sal + sal*0.05 - sal*0.07;
07     printf("Salário a receber: %.2f.\n", liquido);
08 }
```

- 7) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba o salário-base de um funcionário, calcule e mostre o seu salário a receber, sabendo-se que esse funcionário teve gratificação de R\$ 600,00 e paga imposto de 10% sobre o salário base.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída - `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada - `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 7 – Caso de teste

Insira o salário base:

1000

Salário a receber: 1500

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURA SEQUENCIAL – Exercício 7 – Solução

```
01 #include <stdio.h>
02 int main(){
03     float sal, liquido;
04     printf("Insira o salário base:\n");
05     scanf("%f", &sal);
06     liquido = sal + 600.0 - sal*0.1;
07     printf("Salário a receber: %.2f.\n", liquido);
08 }
```

- 8) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba o valor de um depósito e o valor da taxa de juros, calcule e mostre o valor do rendimento e o valor total depois do rendimento.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída - `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada - `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 8 – Caso de teste

```
Insira o valor do depósito:
10000
Qual é a taxa?
15
Rendimento: 1500
Valor total: 11500
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURA SEQUENCIAL – Exercício 8 – Solução

```
01 #include <stdio.h>
02 int main(){
03     float dep, taxa, rend, total;
04     printf("Insira o valor do depósito:\n");
05     scanf("%f", &dep);
06     printf("Qual é a taxa?\n");
07     scanf("%f", &taxa);
08     rend = dep*taxa/100.0;
09     total = dep + rend;
10     printf("Rendimento: %.2f.\n", rend);
11     printf("Valor total: %.2f.\n", total);
12 }
```

- 9) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que calcule e mostre a área de um triângulo. Sabe-se que: $\text{Área} = (\text{base} * \text{altura})/2$.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída - `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada - `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 9 - Caso de teste

```
Insira a base:
4
Insira a altura:
10
Área do triângulo: 20
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURA SEQUENCIAL – Exercício 9 – Solução

```
01 #include <stdio.h>
02 int main(){
03     float base, altura, area;
04     printf("Insira a base:\n");
05     scanf("%f", &base);
06     printf("Insira a altura:\n");
07     scanf("%f", &altura);
08     area = (base * altura)/2;
09     printf("Área do triângulo: %f.\n", area);
10 }
```

- 10) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que calcule e mostre a área de um círculo. Sabe-se que: $\text{Área} = \text{Pi} * R^2$, aonde $\text{Pi} = 3,14$.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída - `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada - `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 10 - Caso de teste

```
Insira o raio:
10
Área do círculo: 314
```


Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURA SEQUENCIAL – Exercício 10 – Solução

```
01 #include <stdio.h>
02 #define P 3.14
03 int main(){
04     float raio, area;
05     printf("Insira o raio:\n");
06     scanf("%f", &raio);
07     area = P * raio * raio;
08     printf("Área do círculo: %f.\n", area);
09 }
```

- 11) Jeremias possui um cronômetro que consegue marcar o tempo apenas em segundos. Sabendo disso, desenvolva um algoritmo que receba o tempo cronometrado, em segundos, e diga quantas horas, minutos e segundos se passaram a partir do tempo cronometrado.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída - `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada - `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 11 – Caso de teste

```
Insira tempo em segundos:
3661
Horas:  1
Minutos: 1
Segundos: 1
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURA SEQUENCIAL – Exercício 11 – Solução

```
01 #include <stdio.h>
02 int main(){
03     int seg, min, hr;
04     printf("Insira o tempo em segundos:\n");
05     scanf("%d", &seg);
06     hr = seg / 3600;
07     seg = seg % 3600;
08     min = seg / 60;
09     seg = seg % 60;
10     printf("Horas: %d.\n", hr);
11     printf("Minutos: %d.\n", min);
```

```
12     printf("Segundos: %d.\n", seg);  
13 }
```

- 12)** Desenvolva um algoritmo que emule um caixa eletrônico. O usuário deve inserir o valor total a ser sacado da máquina e o algoritmo deve informar quantas notas de 100, 50, 20, 10, 5 ou 2 reais serão entregues. Deve-se escolher as notas para que o usuário receba o menor número de notas possível.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída - `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada - `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 12 - Caso de teste

Insira o valor a sacar:

187

nº notas R\$ 100,00: 1

nº notas R\$ 50,00: 1

nº notas R\$ 20,00: 1

nº notas R\$ 10,00: 1

nº notas R\$ 5,00: 1

nº notas R\$ 2,00: 1

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURA SEQUENCIAL – Exercício 12 – Solução

```
01 #include <stdio.h>  
02 int main(){  
03     int cem, cinq, vinte, dez, cinco, dois, saque;  
04     printf("Insira o valor a sacar:\n");  
05     scanf("%d", &saque);  
06     cem = saque / 100;  
07     saque = saque % 100;  
08     cinq = saque / 50;  
09     saque = saque % 50;  
10     vinte = saque / 20;  
11     saque = saque % 20;  
12     dez = saque / 10;  
13     saque = saque % 10;  
14     cinco = saque / 5;  
15     saque = saque % 5;  
16     dois = saque / 2;  
17     printf("nº notas R$ 100,00: %d.\n", cem);  
18     printf("nº notas R$ 50,00: %d.\n", cinq);  
19     printf("nº notas R$ 20,00: %d.\n", vinte);  
20     printf("nº notas R$ 10,00: %d.\n", dez);
```

```
21     printf("nº notas R$ 5,00: %d.\n", cinco);  
22     printf("nº notas R$ 2,00: %d.\n", dois);  
23 }
```

13) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba um número positivo e maior que zero, calcule e mostre:

- a) o número digitado ao quadrado;
- b) o número digitado ao cubo;
- c) a raiz quadrada do número digitado;
- d) a raiz cúbica do número digitado.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída - `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada - `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 13 - Caso de teste

```
Insira um número:  
2  
Quadrado: 4  
Cubo: 8  
Raiz Quadrada: 1.4142135623731  
Raiz Cúbica: 1.25992104989487
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURA SEQUENCIAL – Exercício 13 – Solução

```
01 #include <stdio.h>  
02 #include <math.h>  
03 int main(){  
04     float num, qua, cub, rquad, rcub;  
05     printf("Insira um número:\n");  
06     scanf("%f", &num);  
07     qua = pow(num, 2);  
08     cub = pow(num, 3);  
09     rquad = sqrt(qua);  
10     rcub = pow(num, 1.0/3.0);  
11     printf("Quadrado: %f.\n", qua);  
12     printf("Cubo: %f.\n", cub);  
13     printf("Raiz quadrada: %f.\n", rquad);  
14     printf("Raiz cúbica: %f.\n", rcub);  
15 }
```

- 14)** (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba dois números maiores que zero, calcule e mostre um elevado ao outro.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída - `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada - `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 14 – Caso de teste

```
Insira o primeiro número:
2
Insira o segundo número:
3
Resultado: 8
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURA SEQUENCIAL – Exercício 14 – Solução

```
01 #include <stdio.h>
02 #include <math.h>
03 int main() {
04     float A, B, potencia;
05     printf("Insira o primeiro número:\n");
06     scanf("%f", &A);
07     printf("Insira o segundo número:\n");
08     scanf("%f", &B);
09     potencia = pow(A, B);
10     printf("Resultado: %f.\n", potencia);
11 }
```

- 15)** (Adaptado de ASCENCIO e CAMPOS, 2008) Sabe-se que:

1 pé = 12 polegadas;
1 jarda = 3 pés;
1 milha = 1760 jardas;

Faça um programa que receba uma medida em pés, faça as conversões a seguir e mostre os resultados.

- a) polegadas;
- b) jardas;
- c) milhas.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída - `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada - `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 15 – Caso de teste

```
Insira a medida em pés:  
5280  
Polegadas: 63360  
Jardas: 1760  
Milhas: 1
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURA SEQUENCIAL – Exercício 15 – Solução

```
01 #include <stdio.h>  
02 int main(){  
03     float pes, jar, mil, pol;  
04     printf("Insira a medida em pés:\n");  
05     scanf("%f", &pes);  
06     pol = pes*12.0;  
07     jar = pes/3.0;  
08     mil = jar/1760.0;  
09     printf("Polegadas: %f.\n", pol);  
10     printf("Jardas: %f.\n", jar);  
11     printf("Milhas: %f.\n", mil);  
12 }
```

16) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba o ano de nascimento de uma pessoa e ano atual, calcule e mostre:

- a) a idade dessa pessoa;
- b) quantos anos essa pessoa terá em 2030;

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída - `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada - `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 16 – Caso de teste

```
Insira o ano de nascimento:  
1988  
Insira o ano atual:  
2020
```

```
Idade atual: 32
Idade em 2030: 42
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURA SEQUENCIAL – Exercício 16 – Solução

```
01 #include <stdio.h>
02 int main(){
03     int nasc, atual;
04     printf("Insira o ano de nascimento:\n");
05     scanf("%d", &nasc);
06     printf("Insira o ano atual:\n");
07     scanf("%d", &atual);
08     printf("Idade atual: %d.\n", atual-nasc);
09     printf("Idade em 2030: %d.\n", 2030-nasc);
10 }
```