

INTRODUÇÃO À PROGRAMAÇÃO

Prof.^a Priscilla Abreu

priscilla.braz@rj.senac.br



Introdução à Programação



Roteiro de Aula

- Objetivo da aula
- Correção das atividades de funções e procedimentos

ATIVIDADES

Introdução à Programação



Atividade 1

Para formar um triângulo, algumas condições precisam ser satisfeitas com relação às medidas dos lados dessa figura geométrica. Nesse contexto, faça um programa que leia três valores inteiros positivos, representando possíveis medidas de lados de um triângulo. O programa deve verificar se os valores, de fato, representam medidas válidas para um triângulo e, em caso positivo, exibir que tipo de triângulo foi formado (equilátero, escaleno ou isósceles).

Para isso, você deve implementar duas sub-rotinas (procedimento ou função):

- **validaTriangulo**: função/procedimento que recebe por parâmetro os três valores e retorna se eles formam (ou não) um triângulo. Para formar um triângulo, a medida de cada lado deve ser inferior ou igual a soma das medidas dos outros dois lados.

Exemplos:

- $a = 3, b = 6, c = 7$ formam um triângulo, pois $3 \leq 6+7$ e $6 \leq 3+7$ e $7 \leq 3+6$
- $a = 1, b = 2, c = 9$ não formam um triângulo, pois 9 não é menor ou igual a $1 + 2$
- **tipoTriangulo**: função/procedimento que recebe por parâmetro as medidas dos lados e informa que tipo de triângulo representa.
 - Triângulo equilátero: possui todos os lados iguais;
 - Triângulo escaleno: possui todos os lados diferentes;
 - Triângulo isósceles: possui apenas dois lados iguais.

O programa deve usar apenas variáveis locais!

Introdução à Programação



ATIVIDADE 1 – SOLUÇÃO

```
#include <stdio.h>

int validaTriangulo(int a, int b, int c){
    if((a <= (b + c)) && (b <= (a + c)) && (c <= (b + a))){
        return 1;
    }
    else{
        return 0;
    }
}
```

Introdução à Programação



ATIVIDADE 1 – SOLUÇÃO

```
void tipoTriangulo(int a, int b, int c){  
    if(a == b && b == c){  
        printf("\nTriangulo equilatero!\n");  
    }  
    else{  
        if(a!=b && a!=c && b!=c){  
            printf("Triangulo escaleno!\n");  
        }  
        else{  
            printf("\nTriangulo isosceles!\n");  
        }  
    }  
}
```

Introdução à Programação



ATIVIDADE 1 – SOLUÇÃO

```
int main(){
    int a, b, c;
    printf("Informe a medida do primeiro possível lado: ");
    scanf("%d",&a);
    printf("Informe a medida do segundo possível lado: ");
    scanf("%d",&b);
    printf("Informe a medida do terceiro possível lado: ");
    scanf("%d",&c);
    if(validaTriangulo(a,b,c) == 1){
        printf("\nMedidas formam um triangulo!\n");
        tipoTriangulo(a,b,c);
    }
    else
        printf("\nMedidas não formam um triangulo!");
}
```

Introdução à Programação



Atividade 2

As camadas atmosféricas da Terra estão definidas na tabela abaixo.

Camada	Distância (em metros)	
	Mínima	Máxima
troposfera	0	12.000
estratosfera	12.001	50.000
mesosfera	50.001	80.000
termosfera	80.001	500.000
exosfera	500.001	800.000

Faça um programa que leia a altura (distância) de uma nave espacial (em metros), sua velocidade de descida (em metros por segundo) e exiba o nome da camada atmosférica em que ela se encontra e em quanto tempo ela atingirá o solo, mantendo a velocidade de descida constante. O seu programa deverá, obrigatoriamente, implementar e usar as seguintes funções:

- Um procedimento que recebe como parâmetro uma altura (distância) em metros e exibe o nome da camada atmosférica em que ela se encontra. Seu protótipo é: void msg_camada(float altura);
- Uma função que recebe como parâmetros uma altura(distância) e uma velocidade de descida de uma nave espacial (em metros por segundo) e retorna o tempo em segundos para a nave atingir o solo. Seu protótipo é: float tmp_aterrissagem(float altura, float velocidade);
Observação: tempo = distância / velocidade.

Introdução à Programação



ATIVIDADE 2 – SOLUÇÃO

```
#include <stdio.h>
float tmp_aterrisagem (float altura, float velocidade){
    return (altura/velocidade);
}
```

Introdução à Programação



ATIVIDADE 2 – SOLUÇÃO

```
void msg_camada (float altura){  
    if(altura>=0 && altura <=12000){  
        printf("\nCamada: Troposfera\n");  
    }else{  
        if(altura>=12001 && altura<=50000){  
            printf("\nCamada: Estratosfera\n");  
        }  
        else{  
            if(altura>=50001 && altura<=80000){  
                printf("\nCamada: Mesosfera\n");  
            }  
            else{  
                if(altura>=80001 && altura<=500000){  
                    printf("\nCamada: Termosfera\n");  
                }  
            }  
        }  
    }  
}
```

Introdução à Programação



ATIVIDADE 2 – SOLUÇÃO

```
else{
    if(altura>=500001 && altura<=800000){
        printf("\nCamada: Exosfera\n");
    }
    else{
        printf("\nCamada inválida!\n");
    }
}
}
}
}
}
```

Introdução à Programação



ATIVIDADE 2 – SOLUÇÃO

```
int main()
{
    float altura, velocidade;
    printf("Informe a altura/distancia em que se encontra a nave: ");
    scanf("%f",&altura);
    printf("Informe a velocidade da nave: ");
    scanf("%f",&velocidade);
    msg_camada(altura);
    printf("\nTempo      para      atingir      o      solo:      %.1f",
tmp_aterrisagem(altura,velocidade));
    return 0;
}
```

Introdução à Programação



Atividade 3

Uma empresa do setor de alimentos resolveu conceder uma gratificação de Natal aos seus funcionários, com base no número de horas extras e no número de horas que o funcionário faltou ao trabalho. O valor do prêmio é obtido pela consulta à tabela que se segue, na qual:

$$H = \text{nº de horas extras} - (1/3 * (\text{numero_de_horas_falta}))$$

H	Prêmio (R\$)
> 20	500,00
<u>>= 10</u> e < 20	400,00
>= 5 e < 10	300,00
< 5	200,00

Com base nessa situação, faça um programa que **leia o número de horas extras e o número de horas** que o funcionário faltou e informe o valor da gratificação recebida. O programa deve usar apenas variáveis locais.

Para isso, você deve fazer as seguintes funções:

- **calculaH**: essa função deve receber por parâmetro o número de horas extras e o número de horas de faltas e calcular e retornar o valor de H, como apresentado na fórmula.
- **verificaGratificacao**: função que recebe por parâmetro o valor de H e retorna o valor da gratificação a ser recebida pelo funcionário, de acordo com a tabela apresentada.

Introdução à Programação



ATIVIDADE 3 – SOLUÇÃO

```
#include <stdio.h>
float calculaH(float numHE, float numHF){
    return (numHE - (numHF/3));
}
```

Introdução à Programação



ATIVIDADE 3 – SOLUÇÃO

```
float verificaGratificacao(float h){  
    if( h < 5){  
        return 200;  
    }  
    else{//h>=5  
        if(h>=5 && h < 10){  
            return 300;  
        }  
        else{  
            if(h>=10 && h <20){  
                return 400;  
            }  
            else{  
                return 500;  
            }  
        }  
    }  
}
```

Introdução à Programação



ATIVIDADE 3 – SOLUÇÃO

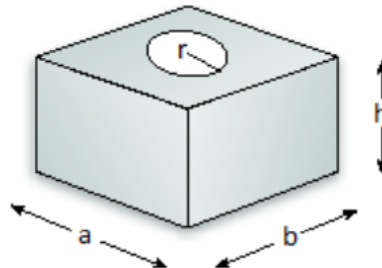
```
int main()
{
    float numHExtras, numHFaltas;
    printf("Informe o numeros de horas extras: ");
    scanf("%f",&numHExtras);
    printf("Informe o número de faltas em horas: ");
    scanf("%f",&numHFaltas);
    printf("\nValor          da          gratificação:          %.2f",
verificaGratificacao(calculaH(numHExtras, numHFaltas)));
    return 0;
}
```


Atividade 4

Sabe-se que o volume de uma caixa de lados **a** e **b** e altura **h** é dado por $V_{caixa} = a * b * h$ e que o volume de um cilindro de raio **r** e altura **h** é dado por $V_{cilindro} = \pi * h * r^2$.

Nesse contexto, faça um programa para calcular o volume de uma caixa com **um furo cilíndrico**, com as dimensões ilustradas na figura abaixo. O programa deve ler os valores das dimensões da peça (**a**, **b**, **h** e **r** – todos reais) e exibir o volume calculado, implementando e utilizando as seguintes funções:

- a) uma função volumeCaixa para calcular e retornar o volume de uma caixa de lados **a** e **b** e altura **h**. Esta função deverá receber os valores de **a**, **b** e **h** como parâmetros;
- b) uma função volumeCilindro para calcular e retornar o volume de um cilindro de raio **r** e altura **h**. Esta função deverá receber o raio e a altura como parâmetros.



Introdução à Programação



ATIVIDADE 4 – SOLUÇÃO

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

```
#define pi 3.14
```

```
float volumeCaixa(float a, float b, float h){  
    return (a*b*h);  
}
```

```
float volumeCilindro(float h, float r){  
    return (pi * h * r * r);  
}
```

Introdução à Programação



ATIVIDADE 4 – SOLUÇÃO

```
int main() {  
    float a, b, h, r, volFinal;  
    printf("Informe as medidas da caixa: ");  
    printf("\nLargura: ");  
    scanf("%f",&a);  
    printf("\nComprimento: ");  
    scanf("%f",&b);  
    printf("\nAltura: ");  
    scanf("%f",&h);  
    printf("\nInforme o raio da base do cilindro: ");  
    scanf("%f",&r);  
    volFinal = volumeCaixa(a, b, h) - volumeCilindro(h, r);  
    printf("\nVolume final: %.2f", volFinal);  
}
```