

Ficha de Trabalho N.º 1 – Soluções Propostas

1 - Elabore o algoritmo e implemente um programa que calcule a área de um triângulo retângulo.

Algoritmo:

Entradas: base, altura;

Saídas: Area.

Variáveis: base, altura, area : REAL;

INICIO

```
ESCREVER('Insira valor da base: ');  
LER (base);  
ESCREVER('Insira valor da altura: ');  
LER (altura);  
area ← base*altura/2;  
ESCREVER('A área do triângulo é ', area);  
{As duas linhas anteriores poderiam ser  
substituídas por uma única, como se segue:  
Escrever ('A área do triângulo é ', base*altura/2);  
}
```

FIM.

Programa:

```
//Exercício 1  
#include <stdio.h>  
#include <locale.h>  
int main(void)  
{  
    setlocale(LC_ALL, "Portuguese");  
    //Exercício 1  
    float base, altura, area;  
    printf("Qual o valor da base do triângulo? ");  
    scanf("%f", &base);  
    printf("Qual o valor da altura do triângulo? ");  
    scanf("%f", &altura);  
    area = base*altura / 2;  
    printf("A área do triângulo vale %.2f\n", area);  
    return 0;  
}
```

2 - Dado o preço de um determinado produto e uma quantidade de dinheiro disponível, elabore o algoritmo e implemente o programa que permita determinar quantas unidades desse artigo pode comprar e quanto dinheiro sobra.

Algoritmo:

Entradas: preço, dinheiro;

Saídas: unidades, troco.

VARIAVEIS preço, dinheiro, unidades, troco : INTEIRO;

INICIO

```
ESCREVER('Qual o preço do Artigo? ');  
LER (preço);  
ESCREVER('Qual o dinheiro disponível?');  
LER (dinheiro);  
unidades ← dinheiro DIVISÃO_INTEIRA preço;  
troco ← dinheiro - unidades *preço;  
ESCREVER('Com ',dinheiro,' pode comprar ');  
ESCREVER(unidades,' produtos e ainda receber ',troco);
```

FIM.

Programa:

```
//Exercício 2
#include <stdio.h>
#include <locale.h>
int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    float dinhDisp, precoUnit, troco;
    int unidades;
    printf("Quanto dinheiro tem disponível? ");
    scanf("%f", &dinhDisp);
    printf("Qual o preço unitário? ");
    scanf("%f", &precoUnit);
    unidades = dinhDisp / precoUnit;
    troco = dinhDisp - precoUnit*unidades;
    printf("Com %.2f euros, compra %d unidades e recebe %.2f de troco\n",
           dinhDisp, unidades, troco);

    return 0;
}
```

3 - Escreva um algoritmo e elabore o programa que permita determinar o maior de três números inteiros dados.

Algoritmo:

Entradas: x1, x2, x3;

Saídas: maior.

```
VARIÁVEIS x1, x2, x3, maior: INTEIRO;
INICIO
    ESCREVER('Insira os valores de x1, x2 e x3: ');
    LER (x1, x2, x3);
    SE x1 > x2 ENTÃO
        maior ← x1;
    SENÃO
        maior ← x2;
    FIM-SE
    SE x3 > maior ENTÃO
        maior ← x3;
    FIM-SE
    ESCREVER('O maior valor é ', maior);
FIM
```

Programa:

```
//Exercício 3
#include <stdio.h>
#include <locale.h>
int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    int num1, num2, num3, maior;
    printf("Insira 3 números inteiros: ");
    scanf("%d%d%d", &num1, &num2, &num3);
    if (num1 >= num2)
        maior = num1;
    else
        maior = num2;
    //definiu-se maior entre num1 e num2
    if (num3 > maior)
    {
        maior = num3;
    }
    printf("O maior vale %d\n", maior);
    return 0;
}
```

4 - Elabore um programa em linguagem C que leia um número inteiro e verifique se é ou não positivo.

Algoritmo:

Apesar do algoritmo não ser solicitado, apresenta-se, para facilitar o trabalho dos estudantes.

Entradas: n;

Saídas: resposta no ecrã, informando se o número especificado é positivo ou negativo.

```
VARIÁVEIS n: INTEIRO;
INICIO
    ESCREVER('Escreva um numero inteiro: ');
    LER (n);
    SE n > 0 ENTÃO
        ESCREVER('O número inserido ', n, ' é positivo!');
    SENÃO
        ESCREVER('O número inserido ', n, ' é negativo!');
    FIM-SE
    SE x3 > maior ENTÃO
        maior ← x3;
    FIM-SE
    ESCREVER('O maior valor é ', maior);
FIM
```

Programa:

```
#include <stdio.h>
#include <locale.h>
int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    int n;
    printf("\nEscreva um numero inteiro: ");
    scanf("%d", &n);
    if (n>0)
        printf("\n %d e positivo!\n", n);
    else
        printf("\n %d nao é positivo!\n", n);
    return 0;
}
```

5 - Elabore um programa em linguagem C que leia um número inteiro e verifique se é par (exercício 3 da antiga ficha 2).

Programa:

```
#include <stdio.h>
#include <locale.h>
int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    int x;

    printf("\nQual o valor de x? ");
    scanf("%d", &x);
    if (x%2 != 0)
        printf("\nx é ímpar!");
    else
        printf("\nx é par!");
    return 0;
}
```

ou:

```
if (x%2)
    printf("\nx é ímpar!");
else
    printf("\nx é par!");
```

6 - Elabore o algoritmo e implemente o programa que determine o terceiro lado de um triângulo retângulo, dados os outros dois (exercício 6 da antiga ficha 1).

Algoritmo:

Entradas: lado1, lado2, qual;

Saídas: lado3.

Teorema de Pitágoras: $a^2 = b^2 + c^2$: Em qualquer triângulo retângulo, o quadrado do comprimento da hipotenusa é igual à soma dos quadrados dos comprimentos dos catetos.

```
VARIAVEIS  lado1, lado2, lado3: REAL;
           qual:INTEIRO;

INICIO
    ESCREVER('Qual o lado que falta? ');
    ESCREVER('[1: Hipotenusa 2: Um cateto]');
    LER (qual);
    SE qual = 1 ENTÃO
        INICIO
            ESCREVER ('Insira valor do cateto 1: ');
            LER (lado1);
            ESCREVER ('Insira valor do cateto 2: ');
            LER (lado2);
            Lado3 ← RaizQ((Quadrado(lado1)+ Quadrado(lado2)));
            ESCREVER ('O valor da hipotenusa é ', lado3);
        FIM
    SENÃO
        INICIO
            ESCREVER ('Insira valor do cateto 1: ');
            LER (lado1);
            ESCREVER ('Insira valor da hipotenusa: ');
            LER (lado2);
            Lado3 ← Quadrado(lado2)- Quadrado(lado1);
            SE Lado3 >0 ENTÃO
                ESCREVER ('O valor do lado é ', RaizQ(lado3));
            SENÃO
                ESCREVER ('Triângulo impossível');
            FIM-SE
        FIM
    FIM-SE
FIM.
```

Programa:

```
// Exercício 6 - Elabore o algoritmo de um programa que determine o terceiro lado de um
// triângulo rectângulo, dados os outros dois.
#include <stdio.h>
#include<stdlib.h>
#include<math.h>
#define _CRT_SECURE_NO_WARNINGS
#include <locale.h>
int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    float lado1=0, lado2=0, lado3=0;
    int ladoQueFalta=0;
    printf("Qual o lado que falta? ");
    printf("[1 - Hipotenusa; 2 - Cateto]: ");
    scanf("%d",&ladoQueFalta);
    printf("\n");
    if (ladoQueFalta==1)
    {
        printf("Insira o valor do cateto 1: ");
        scanf("%f", &lado1);
```

```

    printf("Insira o valor do cateto 2: ");
    scanf("%f", &lado2);
    lado3=sqrt(lado1*lado1+lado2*lado2);
    printf("O valor da hipotenusa: %5.2f", lado3);
}
else
    if (ladoQueFalta==2)
    {
        printf("Insira o valor do cateto 1: ");
        scanf("%f", &lado1);
        printf("Insira o valor da hipotenusa: ");
        scanf("%f", &lado2);
        lado3=sqrt(lado1*lado1-lado2*lado2);
        if (lado3 >0)
            printf("O valor do cateto: %5.2f", lado3);
        else
            printf("Triangulo Impossivel!!");
    }
    else
        printf("Tipo de calculo invalido!!");
printf("\n\n");
return 0;
}

```

7 - Escreva um programa que leia dois números a e b (inteiros) e verifique se a é múltiplo de b ou se b é múltiplo de a.

Programa:

```

//Exercício 7
#include <stdio.h>
#include <stdio.h>
#include <locale.h>
int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    int a, b;
    printf("Introduza dois inteiros: ");
    scanf("%d%d", &a, &b);
    if (a % b == 0)
        printf("%d é múltiplo de %d\n", a, b);
    else
    {
        if (b % a == 0)
            printf("%d É múltiplo de %d\n", b, a);
        else
            printf("Os números não são múltiplos um do outro\n");
    }
    return 0;
}

```

8 - Elabore o algoritmo e implemente o programa que permita calcular a área e o perímetro das seguintes figuras geométricas: quadrado, retângulo e círculo (exercício 7 da antiga ficha 1).

Algoritmo:

Entradas: opcao, ladoA, ladoB, raio;

Saídas: area, perimetro.

CONSTANTES pi=3.14159;

VARIAVEIS ladoA, ladoB, raio, área, perimetro: REAL;

opcao: CHARACTER;

INICIO

```

    ESCRIVER('Calcular área e perímetro de: ');
    ESCRIVER('Q - Quadrado; R - Rectângulo ; C - Círculo');
    LER(opcao);
    CASO opcao SEJA
        'Q', 'q': ESCRIVER('Qual a dimensão do lado?');

```

```

LER (ladoA);
area ← ladoA*ladoA;
perimetro ← 4*ladoA;
ESCREVER ('A area é ',area,' e o perimetro', perimetro);
'R', 'r': ESCRIVER('Quais as dimensões dos lados?');
LER (ladoA,ladoB);
area ← ladoA*ladoB;
perimetro ← 2*(ladoA+ladoB);
ESCREVER ('A area é ',area,' e o perimetro', perimetro);
'C', 'c': ESCRIVER('Qual o raio do círculo?');
LER (raio);
area ← pi*raio*raio;
perimetro ← 2*pi *raio;
ESCREVER ('A area é ',area,' e o perimetro', perimetro);
OUTROS : ESCRIVER ('Opção inválida');
FIM-CASO
FIM.

```

Programa:

```

// Exercício 8
// Elabore o algoritmo de um programa que permita calcular a área e o
// perímetro das seguintes figuras geométricas: quadrado, rectângulo e círculo.
#include <stdio.h>
#include <locale.h>
#define _CRT_SECURE_NO_WARNINGS
int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    const double pi=3.14159;
    float ladoA, ladoB, raio, area, perimetro;
    char opcao='Q';
    printf("*** Programa que permite o cálculo da área e perimetro de várias figuras
                                geométricas ***\n\n");
    printf("Calcular a área e perímetro de ");
    printf("Q – Quadrado; R – Rectângulo ; C – Círculo: ");
    scanf("%c", &opcao);
    switch (opcao)
    {
        case 'Q':
        case 'q':
            printf("Qual a dimensão do lado? ");
            scanf("%f", &ladoA);
            area = ladoA*ladoA;
            perimetro = 4*ladoA;
            printf("A area do quadrado = %.2f e o perimetro = %.2f\n\n", area, perimetro);
            break;
        case 'R':
        case 'r':
            printf("Quais as dimensões dos lados (separe os valores por um espaço) ? ");
            scanf("%f %f", &ladoA, &ladoB);
            area = ladoA * ladoB;
            perimetro = 2 * (ladoA + ladoB);
            printf("A area do rectangulo = %.2f e o perimetro = %.2f\n\n", area, perimetro);
            break;
        case 'C':
        case 'c':
            printf("Qual o raio do círculo ? ");
            scanf("%f", &raio);
            area = pi * raio * raio;
            perimetro = 2 * pi * raio;
            printf("A area do círculo = %.2f e o perimetro da circunferencia = %.2f\n\n",
                    area, perimetro);
            break;
        default:
            printf("Opcao Invalida!!\n\n");
    }
    return 0;
}

```

- 9 - Elabore um programa que leia dois números e calcule (apresentando os resultados no monitor): a soma, a divisão inteira, o resto da divisão inteira e o produto (exercício 15 da antiga ficha 1).

Programa:

```
// Exercício 6 – Elabore um programa que leia dois números e calcule (apresentando os
// resultados no monitor):
// a soma, a divisão inteira, o resto da divisão inteira e o produto.
#include <stdio.h>
#include <locale.h>
#include <stdio.h>
#include <locale.h>
int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    int x,y;
    printf("Qual o primeiro número? ");
    scanf("%d",&x);
    printf("\nQual o segundo número? ");
    scanf("%d",&y);
    printf("Soma: %d\n", x+y);
    printf("Produto: %d\n", x*y);
    if (y!=0)
    {
        printf("Resultado da divisão inteira: %d\n", x/y);
        printf("Resto da divisão inteira: %d\n", x%y);
    }
    else printf("\n Denominador nulo: divisáo impossível!");
    return 0;
}
```

- 10 - Elabore o algoritmo e implemente o programa que determine se um ano dado é comum ou bissexto.

Note que um ano é bissexto se for divisível por 4 mas não por 100, exceto se for divisível por 400.

Algoritmo:

Entradas: ano;

Saídas: mensagem (ano comum ou bissexto).

VARIÁVEIS ano: INTEIRO;

INICIO

 ESCREVER('Qual o ano?');

 LER(ano);

 SE ((ano MOD 4 = 0) **E** (ano MOD 100 <> 0)) **OU** (ano MOD 400 = 0) ENTÃO
 ESCREVER (ano, ' é ano bissexto');

 SENÃO

 ESCREVER (ano, ' é ano comum');

 FIM-SE

FIM.

Programa:

```
// Exercício 9 –Elabore o algoritmo de um programa que determine se um ano dado é
// comum ou bissexto.
// Note que um ano é bissexto se for divisível por 4 mas não por 100,
// excepto se for divisível por 400.

#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#define _CRT_SECURE_NO_WARNINGS
int main(void)
{
    setlocale(LC_ALL, "Portuguese");+
    int ano=0;
    printf("*** Programa que permite determinar se um ano é comum ou bissexto ***\n\n");
    printf("Qual o ano a testar? ");
```

```

scanf("%d", &ano);

if(((ano % 4 == 0) && (ano % 100 != 0)) || (ano % 400 == 0))
    printf("%d é um ano bissexto.\n", ano);
else
    printf("%d é um ano comum.\n", ano);
return 0;
}

```

11 - Elabore um algoritmo e implemente o programa que permita determinar o número de dias que faltam até ao fim do mês, numa determinada data.

Algoritmo:

Entradas: dia, mês, ano;

Saídas: dias.

```

VARIAVEIS dia, mes, ano, dias: INTEIRO;
INICIO
    ESCRIVER('Escreva uma data (dia, mês, ano)');
    LER(dia, mes, ano);
    CASO mes SEJA
        1,3,5,7,8,10,12: dias ← 31-dia;
        4,6,9,11: dias ← 30-dia;
        2: SE ((ano MOD 4 =0) E (ano MOD 100 <>0)) OU (ano MOD 400 =0) ENTÃO
            dias ← 29-dia;
        SENÃO
            dias ← 28-dia;
        FIM-SE
    OUTROS: ESCRIVER ('Mês inválido');
    FIM-CASO;
    SE (mes>0) E (mes <13) ENTÃO
        ESCRIVER('Faltam ',dias,' dias até ao fim do mês!');
    FIM-SE
FIM.

```

Programa:

```

//Exercício 11
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#define _CRT_SECURE_NO_WARNINGS
int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    int dia, mes, ano;
    printf("Introduza data (dia mes ano): ");
    scanf("%d%d%d", &dia, &mes, &ano);
    if ((ano < 1582) || (ano > 2100))
        printf("Ano inválido!!\n");
    else
    {
        //ano válido, entre 1582 e 2100
        switch (mes)
        {
            case 1:
            case 3:
            case 5:
            case 7:
            case 8:
            case 10:
            case 12:
                if ((dia < 1) || (dia > 31))
                    printf("Dia inválido!\n");
                else //dia est. entre 1 e 31
                    printf("Faltam %d dias para o fim do mês!\n", 31 - dia);
                break;
            case 4:
            case 6:
            case 9:
            case 11:

```



```

        if ((dia < 1) || (dia > 30))
            printf("Dia inválido!\n");
        else
            printf("Faltam %d dias para o fim do mês!\n", 30 - dia);
        break;
case 2:
    if ((ano % 400 == 0) || ((ano % 4 == 0) && (ano % 100 != 0)))
    {
        if ((dia < 1) || (dia > 29))
            printf("Dia inválido!\n");
        else
            printf("Faltam %d dias para o fim do mês!\n", 29 - dia);
    }
    else//não é bissexto
    {
        if ((dia < 1) || (dia > 28))
            printf("Dia inválido!\n");
        else
            printf("Faltam %d dias para o fim do mês!\n", 28 - dia);
    }
    break;
default: printf("Mês inválido!!\n");
        break;
    }
}
return 0;
}

```

12 - Escreva uma instrução de atribuição em C para cada uma das seguintes ações:

- A variável inteira **i** é incrementada uma unidade.
- A variável lógica **v** é verdadeira se e só se a variável inteira **x** tomar o valor **8** ou o valor **80**.
- A variável inteira **r** toma o valor do resto da divisão de **x** por 2.
- A variável lógica **m** é verdadeira se e só se **x** for múltiplo de **n**.
- A variável lógica **maior** é verdadeira se e só se a variável **x** for maior que a variável **y**.

- `i = i + 1;`
- `v = (x == 8) || (x == 80);`
- `r = x % 2;`
- `m = (x % n) == 0;`
- `maior = x > y;`

13 - Elabore um programa que leia o número de minutos decorridos desde a meia-noite e mostre esse número no formato horas:minutos. Por exemplo, se o número lido for 515 deve ser mostrado 8:35, se for 1335 deve ser mostrado 22:15. Tenha em atenção que o dia tem 1440 minutos.

Programa:

```

// Exercício 13
#include <stdio.h>
#include <locale.h>
#define _CRT_SECURE_NO_WARNINGS
int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    int min, horas, minutos;

    printf("\nQuantos minutos decorreram desde a meia-noite?");
    scanf("%d", &min);
    if (min < 0 || min > 1440){
        printf("\nValor inválido, o dia tem 1440 minutos!");
        return 0;
    }
    else
    {

```

```

        horas = min / 60;
        minutos = min % 60;
        printf("\n%d minutos correspondem a %d:%d.\n\n", min, horas, minutos);
    }
    return 1;
}

```

14 - Modifique o programa anterior de modo a usar o formato 12 horas. Por exemplo, se o número lido for 515 deve ser mostrado 8:35 a.m., se o número for 1335 deve ser mostrado 10:15 p.m.

Programa:

```

// Exercício 14
#include <stdio.h>
#include <locale.h>
#include <math.h>
#define _CRT_SECURE_NO_WARNINGS
int main(int argc, const char * argv[])
{
    setlocale(LC_ALL, "Portuguese");
    int min, horas, minutos;
    char letra;
    printf("\nQuantos minutos decorreram desde a meia-noite?");
    scanf("%d", &min);
    if ( min < 0 || min > 1440)
        printf("\nValor inválido: o dia tem entre 0 e 1440 minutos!");
    else
    {
        horas = min / 60;
        minutos = min % 60;
        if(horas < 12)
            letra='A';
        else{
            horas=horas-12;
            letra='P';
        }
        printf("\n%d minutos correspondem a %d:%d %cM\n\n", min, horas, minutos, letra);
    }
    return 0;
}

```

15 - Escreva um programa que determine as raízes reais de uma equação do 2.º grau $ax^2 + bx + c = 0$, em que a, b e c são pedidos ao utilizador. Não se esqueça de prever a hipótese de a equação não ter raízes reais.

Programa:

```

//Exercício 15
#include <stdio.h>
#include <locale.h>
#define _CRT_SECURE_NO_WARNINGS
int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    float a, b, c, raiz1, raiz2;
    printf("Introduza os coeficientes de ax^2+bx+c=0\n");
    scanf("%f%f%f", &a, &b, &c);
    if (b*b - 4 * a*c < 0)
        printf("Equação sem raízes!\n");
    else// É positivo ou nulo, >=0

```

```

{
    if (b*b - 4 * a*c == 0)
    {
        raiz1 = -b / (2 * a);
        printf("A equação tem uma raiz: %.3f\n", raiz1);
    }
    else// não é negativo nem nulo, logo, é positivo (>0)
    {
        raiz1 = (-b - sqrt(b*b - 4 * a*c)) / (2 * a);
        raiz2 = (-b + sqrt(b*b - 4 * a*c)) / (2 * a);
        printf("A equação tem duas raizes: %.3f; %.3f\n", raiz1, raiz2);
    }
}
return 0;
}

```

16 - Escreva um programa em linguagem C que leia um número inteiro e o escreva duas vezes no monitor, mas formatando a sua saída de dois modos diferentes:

- reservando 8 espaços para a sua escrita;
- reservando 8 espaços para a sua escrita, mas alinhando-o à esquerda;
- preenchendo com serros à esquerda;
- idem a c), mas com sinal obrigatório.

Programa:

```

// Exercício 16
#include <stdio.h>
#include <locale.h>
#define _CRT_SECURE_NO_WARNINGS
int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    int n;
    printf("Escreva um número inteiro ");
    scanf("%d", &n);
    printf(" a)inicio|%8d|fim\n", n);
    printf(" b)inicio|%-8d|fim\n", n);
    printf(" c)inicio|%+08d|fim\n", n); // nova alínea c) com 0 à esquerda
    printf(" d)inicio|%+08d|fim\n", n); // nova alínea d) com 0 à esquerda e sinal obrigatório
    return 0;
}

```

17 - Elabore um programa que leia do teclado um número real e o escreva no monitor:

- limitando a 3 o nº de casas decimais;
- limitando a 7 o nº de casas decimais;
- com 4 casas decimais, ocupando 10 posições totais;
- apresentar o número que foi lido, efetuando várias tentativas. É exactamente o que foi introduzido? Tente encontrar uma explicação para o facto.

Programa:

```

// Exercício 17
#include <stdio.h>
#include <locale.h>
#define _CRT_SECURE_NO_WARNINGS
int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    float n;

```

```

printf("Escreva um número real ");
scanf("%f",&n);
printf("a) %.3f -> número com 3 casas decimais \n",n);
printf("b) %.7f -> número com 7 casas decimais \n",n);
printf("c) %10.4f -> número com 4 casas decimais, ocupando 10 posições \n",n);
printf(" %f -> número lido \n",n);
return 0;
}

```

18 - Considere as seguintes correspondências entre unidades de medida:

$x \text{ cm} \Leftrightarrow y \text{ pol}$	$y = x / 2.54$
$x \text{ Kg} \Leftrightarrow y \text{ lbs}$	$y = x / 0.4536$
$x \text{ l} \Leftrightarrow y \text{ gal}$	$y = x / 3.785$
$x^{\circ}\text{C} \Leftrightarrow y^{\circ}\text{F}$	$y = 1.8 x + 32$

Elabore um algoritmo e implemente o programa que permita converter um valor de uma das unidades para outra.

(exercício 4 da antiga ficha 1).

Algoritmo:

```

VARIAVEIS unidade, valor: INTEIRO;
INICIO
    ESCREVER ('Qual a conversão a efectuar ? ');
    ESCREVER ('\[1. cm->pol; 2. Kg->lbs; 3. l->gal; 4. °C->°F ] ');
    LER (unidade);
    ESCREVER ('Insira o valor a converter');
    LER(valor);
    SE unidade = 1 ENTÃO
        ESCREVER ('Valor em polegadas é ', valor/2.54);
    SENÃO
        SE unidade = 2 ENTÃO
            ESCREVER('Valor em lbs é ', valor / 0.4536);
        SENÃO
            SE unidade = 3 ENTÃO
                ESCREVER(' Valor em galões é ', valor/3.7854);
            SENÃO
                ESCREVER('Valor em graus Fahrenheit é ',valor*1.8+32);
        FIM-SE
    FIM-SE
FIM.

```

Programa:

```

//Exercício 18
#include <stdio.h>
#include <locale.h>
#define _CRT_SECURE_NO_WARNINGS
void main(int){
    setlocale(LC_ALL, "Portuguese");
    int opcao;
    float valor, result;
    printf("Escolha uma opção:\n1: cm->pol\n2: kg->lb\n3: l->gal\n4: C->F\n");
    scanf("%d", &opcao);
    switch (opcao)
    {
        case 1:printf("Quantos cm? ");
               scanf("%f", &valor);
               result = valor / 2.54;
               printf("Correspondem a %.2f pol\n", result);
               break;
        case 2:printf("Quantos kg? ");
               scanf("%f", &valor);
    }
}

```

```

        result = valor / 0.4536;
        printf("Correspondem a %.2f lb\n", result);
        break;
    case 3: printf("Quantos lt? ");
            scanf("%f", &valor);
            result = valor / 3.785;
            printf("Correspondem a %.2f gal\n", result);
            break;
    case 4: printf("Quantos C? ");
            scanf("%f", &valor);
            result = valor * 1.8 + 32;
            printf("Correspondem a %.2f F\n", result);
            break;
    default: printf("OpÁ„o inv·lida!!\n");
             break;
}
return 0;
}

```

19 - Elabore um algoritmo e implemente o programa que determine se um caracter dado é:

- letra minúscula;
- letra maiúscula; número inteiro e o escreva
- um dígito;
- caracter de pontuação;
- outro caracter.

Algoritmo:

Entradas: car;

Saídas: mensagem

VARIAVEIS car: CHARACTER;

INICIO

 ESCREVER('Insira um caracter: ');

 LER(car);

 SE car >= 'A' E car <= 'Z' ENTÃO

 ESCREVER('Letra maiúscula');

 SENÃO

 SE car >= 'a' E car <= 'z' ENTÃO

 ESCREVER ('Letra minúscula');

 SENÃO

 SE car >= '0' E car <= 9 ENTÃO

 ESCREVER ('Dígito');

 SENÃO

 SE car='.' OU car=';' OU car=':' OU car='.', (...) ENTÃO

 ESCREVER ('Caracter de pontuação');

 SENÃO

 ESCREVER ('Outro caracter');

 FIM-SE

 FIM-SE

 FIM-SE

 FIM-SE

FIM.

Programa:

```

//Exercício 19
#include <stdio.h>
#include <locale.h>
#define _CRT_SECURE_NO_WARNINGS
int main(void){
    setlocale(LC_ALL, "Portuguese");
    char car;
    printf("Insira caracter: ");
    scanf("%c", &car);
    if ((car >= '0') && (car <= '9'))

```

```

        printf("Dígito!!\n");
    else
    {
        if ((car >= 'A') && (car <= 'Z'))
            printf("Maiúscula!!\n");
        else
        {
            if ((car >= 'a') && (car <= 'z'))
                printf("minúscula!!\n");
            else
            {
                if(car=='!' || car=='?' || car=='.' || car==';' || car==',')
                    printf("Pontuação!!\n");
                else
                    printf("Outro caracter!!\n");
            }
        }
    }
    return 0;
}

```

- 20** - Elabore um algoritmo e implemente o programa para simulação de uma calculadora rudimentar que efetue as quatro operações aritméticas básicas: adição, subtração, multiplicação e divisão.

Algoritmo:

Entradas: operacao, operando1, operando2;

Saídas: resultado.

```

VARIÁVEIS operando1, operando2, resultado: REAL;
operacao: CHARACTER;
INICIO
    ESCRIVER('Qual a operação?');
    LER(operacao);
    ESCRIVER('Quais os dois operadores ?');
    LER(operando1, operando2);
    CASO operacao SEJA
        '+' FAZ : resultado ← operando1 + operando2;
            ESCRIVER ('O resultado da adição é ', resultado);
        '-' FAZ : resultado ← operando1 - operando2;
            ESCRIVER ('O resultado da subtração é ', resultado);
        '*' FAZ : resultado ← operando1 * operando2;
            ESCRIVER ('O resultado da multiplicação é ', resultado);
        ':', '/' FAZ: resultado ← operando1 / operando2;
            ESCRIVER ('O resultado da divisão é ', resultado);
    OUTROS: ESCRIVER ('Operação inválida');
    FIM-CASO
FIM.

```

Programa:

```

//Exercício 20
#include <stdio.h>
#include <locale.h>
#define _CRT_SECURE_NO_WARNINGS
void main(){
    setlocale(LC_ALL, "Portuguese");
    float num1, num2, result;
    char operador;
    printf("Qual a operação +, -, *, / \n");
    scanf("%c", &operador);
    printf("Introduza dois números: ");
    scanf("%f%f", &num1, &num2);
    switch (operador)
    {
        case '+': result=num1+num2;
            printf("A soma vale %.3f\n", result);
            //printf("A soma vale %.3f\n", num1 + num2);

```

```

        break;
    case '-': printf("A diferença vale %.3f\n", num1 - num2);
        break;
    case '*': printf("A multiplicação vale %.3f\n", num1 * num2);
        break;
    case '/':
        if (num2 != 0) //igual de comparação É == -> if(num2==0)
            printf("A divisão vale %.3f\n", num1 / num2);
        else
            printf("Não se pode dividir por 0!\n");
        break;
    default: printf("Operador inválido!!\n");
        break;
}
}

```

21 - Escreva um programa que mostre o tamanho em bytes ocupado por cada tipo de dados numéricos e o respetivo valor mínimo e máximo.

Programa:

```

//Exercício 21
#include <stdio.h>
#include <locale.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <locale.h>
#define _CRT_SECURE_NO_WARNINGS

int main(void){
    setlocale(LC_ALL, "Portuguese");
    printf("*** e respetivo valor mínimo e máximo (para os tipos numéricos inteiros) ***\n\n");
    printf("Tamanho em bytes de um char = %d\n", (int) sizeof(char));
    printf("Tamanho em bytes de um short = %d\n", (int) sizeof(short));
    printf("Para um short int, o valor mínimo = -%d, valor máximo = +%d\n",
        (int)((pow(2.0, (int) (sizeof(short) * 8)) / 2)),
        (int) (pow(2.0, (int) (sizeof(short) * 8))/2) -1);
    printf("Tamanho em bytes de um int = %d\n", (int) sizeof(int));
    printf("Para um int, o valor mínimo = %d, valor máximo = +%d\n",
        (int)((pow(2.0, (int) (sizeof(int) * 8)) / 2)),
        (int) ((pow(2.0, (int) (sizeof(int) * 8))/2) -1);
    printf("Tamanho em bytes de um long = %d\n", (int) sizeof(long));
    printf("Para um long, o valor mínimo = %ld, valor máximo = +%ld\n",
        (long)((pow(2.0, (int) (sizeof(long) * 8)) / 2)),
        (long) ((pow(2.0, (int) (sizeof(long) * 8))/2) -1);
    printf("Tamanho em bytes de um unsigned short int = %d\n", (int) sizeof(unsigned short int));
    printf("Para um unsigned short, o valor mínimo = 0, valor máximo = +%d\n",
        (int)((pow(2.0, (int) (sizeof(unsigned short) * 8))))-1);
    printf("Tamanho em bytes de um unsigned int = %d\n", (int) sizeof(unsigned int));
    printf("Para um unsigned int, o valor mínimo = 0, valor máximo = +%0.f\n",
        (pow(2.0, (int) (sizeof(unsigned int) * 8))))-1;
    printf("Tamanho em bytes de um unsigned long int = %d\n", (int) sizeof(unsigned long int));
    printf("Para um unsigned long, o valor mínimo = 0, valor máximo = +%0.f\n",
        (pow(2.0, (int) (sizeof(unsigned long) * 8))))-1;
    printf("Tamanho em bytes de um float = %d\n", (int) sizeof(float));
    printf("Tamanho em bytes de um double = %d\n", (int) sizeof(double));
    return 0;
}

```