

Para todos os enunciados abaixo, elabore a função/procedimento e faça sua chamada no algoritmo principal.

1. Elabore uma função que verifica se um número é perfeito ou não. Número perfeito é um número natural cuja soma dos seus divisores próprios (excluído o próprio número) coincide com o número.

Número	Divisores	Soma dos divisores
6	1, 2, 3	$1 + 2 + 3 = 6$
28	1, 2, 4, 7, 14	$1 + 2 + 4 + 7 + 14 = 28$
496	1, 2, 4, 8, 16, 31, 62, 124, 248	$1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248 = 496$

```
//Exerc 1
2. #include <stdio.h>

int num_perfeito(int n);

int main()
{
    int valor;

    printf("Digite um número: ");
    scanf("%i", &valor);

    if(num_perfeito(valor) == valor){
        printf("É um número perfeito!");
    }
    else{
        printf("Não é um número perfeito.");
    }
}

int num_perfeito(int n){
    int s = 0;

    for(int i = 2; i <= n; i++){
        if(n % i == 0){
            s += n/i;
        }
    }

    return s;
}
```

Elabore uma função que efetue a exponenciação de um número x por um expoente y (sem usar a função de exponenciação).

Exemplo: $2^{10} \Rightarrow 1024$; $2^{-2} = 1 / 2^2 \Rightarrow 0.25$; $2^0 \Rightarrow 1$

```
//Exerc 2

#include <stdio.h>

float expo(int x, int y);

int main()
{
    int x;
    int y;
    float result;

    printf("Digite um número: ");
    scanf("%i", &x);

    printf("Digite seu expoente: ");
    scanf("%i", &y);

    printf("Resultado: %f", expo(x,y));

    return 0;
}

float expo(int x, int y){
    int e = x;
    if(y == 0){
        return 1;
    }
    else if(y < -1){
        for(int i = -2; i >= y; i--){
            x *= e;
        }
        return 1.f/x;
    }
    else if(y == -1){
        return 1.f/x;
    }
    else if(y == 1){
        return x;
    }
    else if(y > 1){
        for(int i = 2; i <= y; i++){
            x *= e;
        }

        return x;
    }
}
```

faça um algoritmo que leia 3 notas de um aluno e um quarto número que representa o modelo de média a ser calculada. Caso o modelo seja igual a 1, calcular a média aritmética; caso seja igual a 2, calcular a média ponderada com pesos iguais a 5, 3 e 2; caso seja igual a 3, calcular a média harmônica. Repetir este procedimento até que uma das notas informadas seja menor que 0.

$$Pond = \frac{nota\ 1 * peso\ 1 + nota\ 2 * peso\ 2 + nota\ 3 * peso\ 3}{peso\ 1 + peso\ 2 + peso\ 3}$$

$$Harm = \frac{3}{\frac{1}{nota\ 1} + \frac{1}{nota\ 2} + \frac{1}{nota\ 3}}$$

4.

```
//Exerc 3

#include <stdio.h>

float media(float n1, float n2, float n3, int mod);

int main()
{
    float nota1 = 0;
    float nota2 = 0;
    float nota3 = 0;
    int modelo = 0;

    while(nota1 >= 0 && nota2 >= 0 && nota3 >= 0){
        printf("Digite a nota 1: ");
        scanf("%f", &nota1);
        printf("Digite a nota 2: ");
        scanf("%f", &nota2);
        printf("Digite a nota 3: ");
        scanf("%f", &nota3);
        printf("Digite o número do modelo de média: ");
        scanf("%i", &modelo);

        printf("%f \n", media(nota1,nota2,nota3,modelo));
    }

    return 0;
}

float media(float n1, float n2, float n3, int mod){
    if(mod == 1){
        return (n1+n2+n3)/3;
    }
    else if(mod == 2){
        return ((n1*5)+(n2*3)+(n3*2))/9;
    }
    else if(mod == 3){
        return 3/((1.f/n1)+(1.f/n2)+(1.f/n3));
    }
}
```

Você está desenvolvendo um jogo em 3 dimensões no estilo do Counter-Strike. Nesse momento do desenvolvimento é necessário criar uma função de colisão. Essa função deverá receber as coordenadas em 3 dimensões de 2 pontos e a função deve retornar se colidiu ou não (VERDADEIRO ou FALSO). Dessa forma, sua função servirá para as seguintes situações, por exemplo:

- Um tiro acertando um oponente;
- Você está encostando em uma parede;
- Você recebendo um tiro;
- Você pulando e caindo no chão;
- Você recebendo estilhaços de uma granada que explodiu perto;
- E todas as demais situações de colisão do jogo.

Para isso você deve considerar:

- que coordenadas tridimensionais são compostas de 3 informações, dessa forma: (x,y,z);
- a fórmula abaixo para cálculo da distância, ou seja, se o d for 0 é sinal que está colidindo.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

```
//Exerc 4

#include <stdio.h>
#include <stdbool.h>
#include <math.h>

bool colidir(float x1, float y1, float z1, float x2, float y2, float z2);

int main()
{
    float p1x;
    float p1y;
    float p1z;

    float p2x;
    float p2y;
    float p2z;

    printf("(x,y,z) do ponto 1: ");
    scanf("%f %f %f", &p1x, &p1y, &p1z);

    printf("(x,y,z) do ponto 2: ");
    scanf("%f %f %f", &p2x, &p2y, &p2z);

    printf("%s", colidir(p1x,p1y,p1z,p2x,p2y,p2z) == 1 ? "Verdadeiro" : "Falso");

    return 0;
}

bool colidir(float x1, float y1, float z1, float x2, float y2, float z2){
    float d;
    float x;
    float y;
    float z;

    x = pow((x2 - x1),2);
    y = pow((y2 - y1),2);
    z = pow((z2 - z1),2);

    d = sqrt(x + y + z);

    if(d == 0){
        return true;
    }
    else{
        return false;
    }
}
```

5. Desenvolva apenas UM algoritmo que, dependendo da opção que o usuário escolher, execute um dos exercícios abaixo. O programa só deve terminar quando o usuário escolher para sair. Abaixo o exemplo do menu:
- 1 – Exercício 1
 - 2 – Exercício 2
 - 3 – Exercício 3
 - 4 – Exercício 4
 - 5 – Sair

Exercício 1: Crie um procedimento que solicite 3 valores do usuário e chame uma função que necessite de três argumentos, e que retorne a soma do quadrado desses três parâmetros. Imprima esse resultado na tela.

Exercício 2: Crie um procedimento que peça 2 informações ao usuário: linhas e colunas. Ele deve chamar uma função que desenhe um retângulo na tela usando os caracteres '+', '-' e '|'. Esta função deve receber dois parâmetros: linhas e colunas.

Ex.: linhas= 4; colunas = 8

```

+-----+
|       |
|       |
+-----+

```

Obs: para pular linha: ESCRIVA("\n");

Exercício 3: Crie um procedimento que pergunte um número ao usuário e chame uma função que informe a quantidade de dígitos de um determinado número inteiro informado. Imprima quantos dígitos tem esse número.

Exercício 4: Crie procedimento(s) e/ou função(ões), a sua escolha, que solicite um número do usuário (n) e imprima:

```

1
1 2
1 2 3
.. .. ..
1 2 3 .. n

```

Por exemplo:

n = 6

Imprimir:

```

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6

```

```

//Exerc 5

#include <stdio.h>
#include <stdbool.h>
#include <math.h>

int exerc1(int n1, int n2, int n3);
const char* exerc2(int lin, int col);
int exerc3(int x);
int exerc4(int x);

int main()
{
    bool rodar = true;
    int escolha;

    int x;
    int y;
    int z;

    while(rodar){

        printf("Digite uma das opções: \n 1- Exercício 1 \n 2- Exercício 2 \n");
        printf(" 3- Exercício 3 \n 4- Exercício 4 \n 5- Sair \n");
        scanf("%i", &escolha);

        switch (escolha){
            case 1:
                printf("Exercício 1: \n Digite três valores: \n");
                scanf("%i %i %i", &x, &y, &z);
                printf("Resultado: %i \n", exerc1(x,y,z));
                break;

            case 2:
                printf("Exercício 2: \n Digite o número de linhas e colunas: \n");
                scanf("%i %i", &x, &y);
                exerc2(x,y);
                break;

            case 3:
                printf("Exercício 3: \n Digite um número: \n");
                scanf("%i", &x);
                printf("Número total de dígitos: %i \n", exerc3(x));
                break;

            case 4:
                printf("Exercício 4: \n Digite um número: \n");
                scanf("%i", &x);
                exerc4(x);
                break;

            case 5:
                return 0;
                break;
        }
    }

    int exerc1(int n1, int n2, int n3){
        return (pow(n1,2)) + (pow(n2,2)) + (pow(n3,2));
    }

    const char* exerc2(int lin, int col){

```

```

for(int l = 1; l <= lin; l++){
    for(int c = 1; c <= col; c++){
        if((l == 1 || l == lin) && (c == 1 || c == col)){
            if(c == 1){
                printf("+");
            }
            else{
                printf("+\n");
            }
        }
        else if((l == 1 || l == lin) && (c > 1 && c < col)){
            printf("-");
        }
        else if((l > 1 && l < lin) && (c == 1 || c == col)){
            if(c == 1){
                printf("|");
            }
            else{
                printf("| \n");
            }
        }
        else if((l > 1 && l < lin) && (c > 1 && c < col)){
            printf(" ");
        }
    }
}
return 0;
}

int exerc3(int x){
    return log10(x) + 1;
}

int exerc4(int x){
    for(int i = 1; i <= x; i++){
        for(int o = 1; o <= i; o++){
            if(o != i){
                printf("%i ", o);
            }
            else{
                printf("%i \n", o);
            }
        }
    }
}

```