

TRABALHO FUNÇÕES

NOME: KAUÃ ARAUJO DE SOUZA

PASTA DO GOOGLE DRIVE ou GITHUB: [Link Aqui](#)

EXERCÍCIO 6:

```
*****  
*****
```

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

```
// Protótipos das funções
```

```
int fatorial(int n);
```

```
int arranjo(int n, int p);
```

```
int combinacao(int n, int p);
```

```
int main() {
```

```
    int n, p;
```

```
    printf("Cálculo de Arranjo e Combinação:\n");
```

```
    while (1) {
```

```
        // Entrada do usuário para n e p
```

```
        printf("\nDigite os valores de n e p (n >= p >= 0): ");
```

```
        scanf("%d %d", &n, &p);
```

```
        // Validação de n e p
```

```
        if (n < 0 || p < 0 || n < p) {
```

```

        printf("Valores inválidos. Certifique-se de que n >= p >= 0.\n");
        break;
    }

    // Cálculo do arranjo e combinação
    printf("Arranjo A(%d, %d) = %d\n", n, p, arranjo(n, p));
    printf("Combinação C(%d, %d) = %d\n", n, p, combinacao(n, p));
}

return 0;
}

// Função para calcular o fatorial
int fatorial(int n) {
    int resultado = 1, i;
    for (i = 1; i <= n; i++) {
        resultado *= i;
    }
    return resultado;
}

// Função para calcular o arranjo
int arranjo(int n, int p) {
    return fatorial(n) / fatorial(n - p);
}

// Função para calcular a combinação
int combinacao(int n, int p) {

```

```
    return arranjo(n, p) / fatorial(p);  
}
```

```
*****  
*****
```

EXERCÍCIO 9:

```
*****  
*****
```

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

```
// Protótipo da função
```

```
double calcula_imc(char sexo, double massa, double altura);
```

```
int main() {
```

```
    char sexo;
```

```
    double massa, altura, imc;
```

```
    int i;
```

```
    printf("Cálculo do IMC de 100 entrevistados:\n");
```

```
    for (i = 1; i <= 100; i++) {
```

```
        printf("\nEntrevistado %d:\n", i);
```

```
        // Entrada do sexo
```

```
        printf("Digite o sexo (M para Masculino, F para Feminino): ");
```

```
        scanf(" %c", &sexo);
```

```
        // Validação do sexo
```

```
        if (sexo != 'M' && sexo != 'F') {
```

```
        printf("Erro: Sexo inválido. Use 'M' ou 'F'.\n");
        continue;
    }

    // Entrada de massa e altura
    printf("Digite a massa (kg): ");
    scanf("%lf", &massa);

    printf("Digite a altura (m): ");
    scanf("%lf", &altura);

    // Validação da altura
    if (altura <= 0) {
        printf("Erro: Altura deve ser maior que 0.\n");
        continue;
    }

    // Chamada da função para calcular o IMC
    imc = calcula_imc(sexo, massa, altura);

    // Exibição do resultado
    printf("IMC do entrevistado %d: %.2f\n", i, imc);
}

return 0;
}

// Implementação da função
```

```
double calcula_imc(char sexo, double massa, double altura) {
    if (sexo == 'F') {
        return (0.95 * massa) / (altura * altura); // Fórmula para sexo feminino
    } else if (sexo == 'M') {
        return (1.05 * massa) / (altura * altura); // Fórmula para sexo masculino
    }
    return 0; // Valor padrão em caso de erro (não esperado)
}
```

```
*****
*****
```

EXERCÍCIO 11:

```
*****
*****
```

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

```
#include <ctype.h>
```

```
// Protótipo da função
```

```
int classifica_simbolo(char simbolo);
```

```
int main() {
```

```
    char simbolo;
```

```
    int classe, i;
```

```
    printf("Classificação de 100 símbolos:\n");
```

```
    for (i = 1; i <= 100; i++) {
```

```
        printf("\nSímbolo %d:\n", i);
```

```
// Entrada do símbolo

printf("Digite um símbolo: ");

scanf(" %c", &simbolo);


// Chamada da função para classificar o símbolo

classe = classifica_simbolo(simbolo);


// Exibição do resultado

switch (classe) {

    case 1:

        printf("Classe 1: Vocal Minúscula\n");

        break;

    case 2:

        printf("Classe 2: Vocal Maiúscula\n");

        break;

    case 3:

        printf("Classe 3: Consoante Minúscula\n");

        break;

    case 4:

        printf("Classe 4: Consoante Maiúscula\n");

        break;

    case 5:

        printf("Classe 5: Símbolo Especial\n");

        break;

    default:

        printf("Erro na classificação.\n");

        break;

}
```

```

    }
}

return 0;
}

// Implementação da função
int classifica_simbolo(char simbolo) {
    // Verifica se é uma vogal minúscula
    if (simbolo == 'a' || simbolo == 'e' || simbolo == 'i' || simbolo == 'o' || simbolo == 'u')
    {
        return 1;
    }

    // Verifica se é uma vogal maiúscula
    else if (simbolo == 'A' || simbolo == 'E' || simbolo == 'I' || simbolo == 'O' || simbolo
    == 'U') {
        return 2;
    }

    // Verifica se é uma consoante minúscula
    else if (islower(simbolo) && !(simbolo == 'a' || simbolo == 'e' || simbolo == 'i' ||
    simbolo == 'o' || simbolo == 'u')) {
        return 3;
    }

    // Verifica se é uma consoante maiúscula
    else if (isupper(simbolo) && !(simbolo == 'A' || simbolo == 'E' || simbolo == 'I' ||
    simbolo == 'O' || simbolo == 'U')) {
        return 4;
    }

    // Caso contrário, é um símbolo especial

```

```

else {

    return 5;

}

}

*****
*****

```

EXERCÍCIO 12:

```

*****
*****

```

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

```
// Protótipo da função
```

```
void calcula_reajuste(double preco, double porcentagem, double
*preco_reajustado, double *reajuste);
```

```
int main() {
```

```
    double preco, porcentagem, preco_reajustado, reajuste;
```

```
    int i;
```

```
    printf("Cálculo de reajuste para 100 mercadorias:\n");
```

```
    for (i = 1; i <= 100; i++) {
```

```
        printf("\nMercadoria %d:\n", i);
```

```
        // Entrada do preço
```

```
        printf("Digite o preço da mercadoria: ");
```

```
        scanf("%lf", &preco);
```



```
if (preco < 0) {  
    printf("Erro: O preço deve ser um valor positivo.\n");  
    continue;  
}  
  
// Entrada da porcentagem de reajuste  
printf("Digite o percentual de reajuste (em %): ");  
scanf("%lf", &porcentagem);  
  
if (porcentagem < 0) {  
    printf("Erro: O percentual de reajuste deve ser positivo.\n");  
    continue;  
}  
  
// Chamada da função para calcular reajuste  
calcula_reajuste(preco, porcentagem, &preco_reajustado, &reajuste);  
  
// Exibição dos resultados  
printf("Preço original: R$ %.2f\n", preco);  
printf("Reajuste: R$ %.2f\n", reajuste);  
printf("Preço reajustado: R$ %.2f\n", preco_reajustado);  
}  
  
return 0;  
}  
  
// Implementação da função
```

```

void calcula_reajuste(double preco, double porcentagem, double
*preco_reajustado, double *reajuste) {

    *reajuste = (preco * porcentagem) / 100; // Calcula o valor do reajuste em reais

    *preco_reajustado = preco + *reajuste; // Calcula o preço reajustado

}

```

```

*****
*****

```

EXERCÍCIO 17:

```

*****
*****

```

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

```
#include <math.h>
```

```
// Protótipo da função
```

```
void calcula_pg(double a1, double q, int n, double *an, double *sn);
```

```
int main() {
```

```
    double a1, q, an, sn;
```

```
    int n, i;
```

```
    printf("Cálculo de An e Sn para 50 P.G's:\n");
```

```
    for (i = 1; i <= 50; i++) {
```

```
        printf("\nP.G %d:\n", i);
```

```
        // Entrada dos dados
```

```
        printf("Digite o primeiro termo (a1): ");
```

```
        scanf("%lf", &a1);
```

```
printf("Digite a razão (q): ");
scanf("%lf", &q);

printf("Digite a quantidade de termos (n): ");
scanf("%d", &n);

// Validação dos valores
if (n <= 0) {
    printf("Erro: A quantidade de termos deve ser maior que 0.\n");
    continue;
}
if (q == 1) {
    printf("Erro: A razão (q) não pode ser igual a 1 para este cálculo.\n");
    continue;
}

// Chamada da função para calcular An e Sn
calcula_pg(a1, q, n, &an, &sn);

// Exibição dos resultados
printf("Enésimo termo (An): %.2f\n", an);
printf("Soma dos termos (Sn): %.2f\n", sn);
}

return 0;
}
```

```
// Implementação da função

void calcula_pg(double a1, double q, int n, double *an, double *sn) {

    // Cálculo do enésimo termo

    *an = a1 * pow(q, n - 1);

    // Cálculo da soma dos termos

    *sn = a1 * (pow(q, n) - 1) / (q - 1);

}
```

```
*****
*****
```