

## EXERCÍCIO 1 -----

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

int palin(int num) {

    char numChar[6];
    int tamChar,i;
    sprintf (numChar, "%d", num) ; // "imprime num na string numChar

    tamChar = strlen(numChar);

    for (i=0; i < tamChar/2; i++) {
        if (numChar[i] != numChar[tamChar-1-i])
            return 0;
    }
    return 1;
}

int main(void) {
    int num;

    printf("Digite um número: ");
    scanf("%d", &num);

    if (palin(num))
        printf("\n%d é palindromo\n", num);
    else
        printf("\n%d nao é palindromo\n", num);

    return 0;
}
```

## EXERCÍCIO 2 -----

```
#include <stdio.h>

#define NATLETAS 2
#define NNOTAS 6

struct tipoAtleta {
    char nome[40];
    float media;
};

typedef struct tipoAtleta tatleta;

float calculaMedia(float *notas) {
    float maior=-1, menor=11, media=0;
    int i;

    for (i=0; i<NNOTAS; i++) {
        if (notas[i] > maior)
            maior = notas[i];
        if (notas[i] < menor)
            menor = notas[i];
    }
    printf("\nMaior=%.2f Menor=%.2f", maior, menor);
    for (i=0; i<NNOTAS; i++)
        media += notas[i];

    media -= (maior+menor);
    media /= (NNOTAS-2);
    return(media);
}

int main(void) {

    float notas[NNOTAS];
    int i,j;
```

```

tateleata atleta[NATLETAS];

for (i=0; i<NATLETAS; i++) {
    printf("Digite o nome do atleta: ");
    scanf(" %s", atleta[i].nome);

    for (j=0; j<NNOTAS; j++) {
        printf("Digite a nota %d: ", j+1);
        scanf("%f", &notas[j]);
    }

    atleta[i].media = calculaMedia(notas);
    printf("\nMedia=%.2f\n\n", atleta[i].media);

}

printf("\n\n---- RESULTADO ---- ");
for (i=0; i<NATLETAS; i++) {
    printf("\n\nNome: %s", atleta[i].nome);
    printf("\nMédia: %.2f", atleta[i].media);

}
return 0;
}

```

### EXERCÍCIO 3 -----

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

int compara(char *cadeiaA, char *cadeiaB) {
    int posIgual;
    int i, j, tamA, tamB, ocorre=0;
    tamA = strlen(cadeiaA);
    tamB = strlen(cadeiaB);
    posIgual = 0;
    for (i=0; i<tamA; i++) {
        for (j=posIgual; j<tamB; j++) {
            if (cadeiaA[i] == cadeiaB[j]) {
                posIgual = j+1;
                ocorre++;
                break;
            }
        }
        if (ocorre != i+1) //
            return 0;
    }
    return 1;
}

int main(void) {
    char cadeiaA[40], cadeiaB[200];
    int ocorre;
    printf("Digite a cadeia de caracteres A: ");
    fgets(cadeiaA, 20, stdin);
    printf("Digite a cadeia de caracteres B: ");
    fgets(cadeiaB, 200, stdin);
    ocorre = compara(cadeiaA,cadeiaB);
    if (ocorre)
        printf("\n\n> A cadeia A ocorre dentro de B\n");
    else
        printf("\n\n> A cadeia A nao ocorre dentro de B\n");
    return 0;
}
```



#### EXERCÍCIO 4 -----

```
#include <stdio.h>

int mdc(int x, int y) {
    if (y==0)
        return (x);
    else
        return (mdc(y,x%y));
}

int main(void) {

    int x,y,divisor;

    printf("Digite um valor para X: ");
    scanf("%d",&x);
    do {
        printf("Digite um valor para Y (<=X e != 0): ");
        scanf("%d",&y);
    } while (y > x || y == 0);

    divisor = mdc(x,y);
    printf("\n\n>> O máximo divisor comum entre %d e %d é: %d\n", x, y,
divisor);
    return 0;
}
```

## EXERCÍCIO 5 -----

```
#include <stdio.h>

void serie(int i, int j, int k) {
    if (i<=j) {
        printf("%d\n", i);
        serie(i+k,j,k);
    }
}

int main(void) {

    int i,j,k;

    printf("Digite i: ");
    scanf("%d",&i);
    printf("Digite j: ");
    scanf("%d",&j);
    printf("Digite o intervalo k: ");
    scanf("%d",&k);
    serie(i,j,k);

    return 0;
}
```

## EXERCÍCIO 6 -----

```
#include <stdio.h>

int tamanhoPalavra(char *pal) {
    if (*pal=='\0')
        return 0;
    else
        return tamanhoPalavra(pal+1)+1;
}

int main(void) {

    char palavra[20];
    int tam;

    printf("Digite uma palavra: ");
    scanf("%s",palavra);
    tam = tamanhoPalavra(palavra);
    printf("\nO tamanho da palavra é: %d ", tam);
    return 0;
}
```



## EXERCÍCIO 7 -----

```
#include <stdio.h>

int calculaQuoc(int A, int B) {
    if (A < B)
        return 0;
    else
        return calculaQuoc(A-B,B) + 1;
}

int main(void) {

    int A,B,quoc;

    printf("Digite A: ");
    scanf("%d",&A);
    printf("Digite B: ");
    scanf("%d",&B);
    quoc = calculaQuoc(A,B);
    printf("\nO quociente é: %d ", quoc);
    return 0;
}
```

## EXERCÍCIO 8 -----

```
#include <stdio.h>

int main(void) {

    float media, estoque, maior, menor;
    int dias, i;

    printf("Digite a quantidade de dias: ");
    scanf("%d", &dias);
    printf("Digite o estoque do dia 1: ");
    scanf("%f", &estoque);
    maior = estoque;
    menor = estoque;
    media = estoque;
    for (i=2;i<=dias;i++) {
        printf("Digite o estoque do dia %d: ", i);
        scanf("%f", &estoque);
        if (estoque > maior)
            maior = estoque;
        if (estoque < menor)
            menor = estoque;
        media += estoque;
    }
    printf("\n\nO maior estoque é %.2f", maior);
    printf("\n\nO menor estoque é %.2f", menor);
    printf("\n\nA media de estoque é %.2f", media/dias);
    return 0;
}
```

## EXERCÍCIO 9 -----

```
#include <stdio.h>

/* O tamanho da matriz será definido por uma variável Global N.
A tabuada começará do 0, logo para armazenar até a tabuada do número N, a
matriz deve ser de tamanho N+1. */

int N;

void imprimeTabuada(int tabuada[N+1][N+1], int N) {
    int i,j;
    printf("\n\nVocê escolheu a tabuada até %d\n", N);
    for (i=1;i<=N;i++) {
        for (j=1;j<=N;j++)
            printf("%3d ", tabuada[i][j]);
        printf("\n");
    }
}

int main(void) {
    int i,j;

    printf("\nQual será o tamanho da sua tabuada? ");
    scanf("%d", &N);
    int tabuada[N+1][N+1];
    for (i=0;i<=N;i++)
        for (j=0;j<=N;j++)
            tabuada[i][j] = i*j;
    imprimeTabuada(tabuada, N);
    printf("\n\nDigite dois números que queira multiplicar: ");
    scanf("%d %d", &i, &j);
    while (i >= 0 && j >= 0) {
        printf("\n A multiplicação é %d\n\n", tabuada[i][j]);
        printf("Digite dois números que queira multiplicar: ");
        scanf("%d %d", &i, &j);
    }
    printf("\n\nFim do programa!");
    return 0;
}
```

## EXERCÍCIO 10 -----

```
#include <stdio.h>
#define N 3

struct TipoAlunos {
    char nome[40];
    int ano;
    float N1, N2, N3, N4, media;
};
typedef struct TipoAlunos tAlunos;

int main(void) {

    tAlunos vet[N];
    int i;
    float maior;

    for (i=0;i<N;i++) {
        printf("Digite o nome do aluno: ");
        scanf(" %s", vet[i].nome);
        printf("Digite o ano de ingresso do aluno: ");
        scanf("%d", &vet[i].ano);
        printf("Digite as quatro notas do aluno: \n");
        scanf("%f %f %f %f", &vet[i].N1, &vet[i].N2, &vet[i].N3, &vet[i].N4);
        vet[i].media = (vet[i].N1 + vet[i].N2 + vet[i].N3 + vet[i].N4)/4;
        if (vet[i].media > maior)
            maior = vet[i].media;
    }
    printf ("\n\nAlunos com a melhor media (%.2f):\n", maior);
    for (i=0;i<N;i++)
        if (vet[i].media == maior)
            printf("%s \n", vet[i].nome);
    return 0;
}
```

## EXERCÍCIO 11 -----

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

#define N 4

void preencheVetor(float *vet) {
    int i;
    for (i=0;i<N;i++) {
        printf("Digite o numero da posicao %d: ", i);
        scanf("%f", &vet[i]);
    }
}

void mostraVetor(float *vet, char *nomeVet) {
    int i;
    printf("\n\nVETOR %s\n", nomeVet);
    for (i=0;i<N;i++)
        printf("%.1f ", vet[i]);
}

void somaVetor(float *vet1, float *vet2) {
    int i;
    for (i=0;i<N;i++)
        printf("%.1f ", vet1[i]+vet2[i]);
}

void subtraiVetor(float *vet1, float *vet2) {
    int i;
    for (i=0;i<N;i++)
        printf("%.1f ", vet1[i]-vet2[i]);
}

int procuraElemento(float *vet, float num) {
    int i;
```

```

for (i=0;i<N;i++)
    if (num == vet[i])
        return 1;
return 0;
}

int escreveMenu() {
    int op;
    printf("\n\nMENU\n");
    printf("1-soma vetores\n");
    printf("2-subtrai vetores\n");
    printf("3-mostra vetores\n");
    printf("4-procura elemento\n");
    printf("5-finaliza programa\n");
    printf("\n>>>> Escolha uma opção: \n");
    scanf("%d", &op);
    return op;
}

int main(void) {
    float X[N], Y[N], num;
    int op;
    char lixo;

    printf("\n\nPREENCHE VETOR X\n");
    preencheVetor(X);
    printf("\n\nPREENCHE VETOR Y\n");
    preencheVetor(Y);

    do {
        op = escreveMenu();

        switch (op) {
            case 1:
                somaVetor(X,Y);
                break;

```

```

case 2:
    subtraiVetor(X,Y);
    break;
case 3:
    mostraVetor(X, "X");
    mostraVetor(Y, "Y");
    break;
case 4:
    printf("\n\n Digite um elemento para procurar: ");
    scanf("%f", &num);
    if (procuraElemento(X,num))
        printf("\nO elemento pertence a X");
    else
        printf("\nO elemento nao pertence a X");
    if (procuraElemento(Y,num))
        printf("\nO elemento pertence a Y");
    else
        printf("\nO elemento nao pertence a Y");
    break;
case 5:
    printf("\n\n Fim do Programa");
    break;
default:
    printf("\n\n Opcao invalida!");
    break;
}

} while (op != 5);
return 0;
}

```