



SIMULADO – P2

Seus códigos são...

*A box without hinges, key, or lid,
yet golden treasure inside is hid*

*(Uma caixa sem dobradiças, chave ou tampa,
mas dentro há um tesouro dourado escondido)*

– J. R. R. Tolkien, *The Hobbit and Lord of the Rings*

1. Complete o código abaixo nos espaços em branco (nem todos os espaços em branco deverão ser preenchidos) para que ele adicione o nono dígito, ‘9’, no começo de todos os números de celulares. O código deverá receber na primeira linha o número de entradas a serem recebidas, sendo a primeira o nome do cliente, e na próxima linha o número de celular.

Complete o código a seguir nos locais indicados, se necessário

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

typedef char Nome;

typedef struct {
    Nome nomeCompleto [100];
    char celular [15];
} _____;

_____ alterarCelular(_____ c);

int main() {
    int n;

    scanf("%d", &n);
    _____ clientes[n];

    for(int i = 0; i < n; i++)
        scanf("%[^\\n] %s", clientes[i].nomeCompleto, clientes[i].celular);

    for(int i = 0; i < n; i++) {
```

```

        printf("%d-Numero antigo%s\n", i, clientes[i].celular);
        clientes[i] = alterarCelular(clientes[i]);
        printf("%d-Novo numero%s\n", i, clientes[i].celular);
    }

    return 0;
}

----- alterarCelular(----- c) {
    -----
    c.celular[0] = '9';
    return c;
}

```

2. Analise o algoritmo abaixo e/ou realize um teste de mesa a fim de determinar qual será a saída quando a entrada do código for o seu Registro de Aluno (os seis dígitos juntos). Você deve apresentar a impressão no quadro abaixo do código, no mesmo padrão no qual os valores são exibidos a partir do algoritmo.

```

#include <stdio.h>

#define MAX 100
typedef int T;

typedef struct {
    T data[MAX];
    int comeco, fim, qtd_elementos;
} Fila;

void inicia_fila();
void insere_fila(T e);
int fila_vazia();
T remove_fila();

typedef struct {
    T data[MAX];
    int topo;
} Pilha;

void inicia_pilha();
void insere_pilha(T e);
int pilha_vazia();
T remove_pilha();

Fila f;
Pilha p;

int main() {
    inicia_fila();
    inicia_pilha();
    int n;
}

```

```

    for (int i = 0; i < 6; i++) {
        scanf("%1d", &n);
        insere_filha(n);
        insere_pilha(n);
    }

    while (1) {
        if (!fila_vazia()) printf("%d", remove_filha());
        if (!pilha_vazia()) printf("%d", remove_pilha());
        if (fila_vazia() && pilha_vazia()) break;
    }

    printf("\n");

    return 0;
}

void inicia_filha() {
    f.comeco = 0;
    f.fim = -1;
    f.qtd_elementos = 0;
}

void insere_filha(T e) {
    f.qtd_elementos++;
    f.fim = (f.fim + 1) % MAX;
    f.data[f.fim] = e;
}

int fila_vazia() {
    return f.qtd_elementos == 0;
}

T remove_filha() {
    f.qtd_elementos--;
    T temp = f.data[f.comeco];
    f.comeco = (f.comeco + 1) % MAX;
    return temp;
}

void inicia_pilha() {
    p.topo = -1;
}

void insere_pilha(T e) {
    p.data[++(p.topo)] = e;
}

int pilha_vazia() {
    return p.topo == -1;
}

```

```
T remove_pilha() {
    return p.data[(p.topo)--];
}
```

| |
|--|
| |
| |
| |

3. Analise o algoritmo abaixo e/ou realize um teste de mesa a fim de determinar qual será a saída quando a entrada do código for o seu Registro de Aluno (os seis dígitos juntos). Você deve apresentar a impressão no quadro abaixo do código, no mesmo padrão no qual os valores são exibidos a partir do algoritmo.

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

void mis();
void te(int);
int ri();
int o(int);
int treco = 42;

int main() {
    int n;
    mis();
    for (int i = 0; i < 6; i++) {
        scanf("%1d", &n);
        te(ri(n));
        printf("%d_", ri(treco));
        treco = o(n);
    }
    printf("\n");
    return 0;
}

void mis() {
    printf("%d\n", treco++);
    treco %= 9;
}

void te(int tchingo) {
    treco += 3 + tchingo;
}

int ri(int lalala) {
    int treco = 90;
    treco += 12;
    treco = (9 * 2) - treco;
    return -treco - lalala;
}
```

```

}

int o(int stuart) {
    for (int i = 0; i < 10; i++)
        stuart = (stuart * treco) % 10;
    return stuart;
}

```

| |
|--|
| |
| |
| |

4. Complete as funções de manipulação de cadeias de caracteres abaixo nos espaços em branco (nem todos os espaços em branco deverão ser preenchidos) para que fiquem funcionais como o esperado.

```

#include <stdio.h>

#define MAX_NOMES 10
#define MAX_CARACTERES 25

typedef char String;

void receber_string(String s[]);
void copiar_string(String s1[], String s2[]);
void printar_string(String s[]);

int main() {
    int nElementos = 0;
    char listaNomes[MAX_NOMES][MAX_CARACTERES];

    receber_string(listaNomes[nElementos]);
    nElementos++;

    for (int i = 1; i < 10; i++) {
        copiar_string(listaNomes[i], listaNomes[nElementos - 1]);
        nElementos++;
    }

    printf("Quantidade de nomes: %d\n", nElementos);
    for (int i = 0; i < 10; i++)
        printar_string(listaNomes[i]);

    return 0;
}

void receber_string(String s[]) {
    int indice = 0;
    while (scanf(_____, &s[indice]) _____)
        indice++;
}

```

```

    -----
}

void copiar_string(String s1[], String s2[]) {
    int i;
    for (-----)
        -----
}

void printar_string(String s[]) {
    for (-----)
        -----
        printf("\n");
}

```

5. Analise o algoritmo abaixo e/ou realize um teste de mesa a fim de determinar qual será a saída quando a entrada do código for o seu Registro de Aluno (os seis dígitos juntos). Você deve apresentar a impressão no quadro abaixo do código, no mesmo padrão no qual os valores são exibidos a partir do algoritmo.

```

#include <stdio.h>

enum coord {
    X = 0, Y
};

typedef struct {
    int numero;
    int coordenadas[2];
} Local;

void ta(Local casa);
Local mu(Local x);
void dan(int numero);
int doo(int k);
void tu(int coordenadas[]);
void dinho(int coordenadas[], Local e);

int main() {
    Local casa;
    int ra;

    scanf("%2d%2d%2d", &casa.numero, &casa.coordenadas[X], &casa.coordenadas[Y]);

    ta(casa);
    casa = mu(casa);
    dan(casa.numero);
    casa.numero = doo(casa.numero);
    tu(casa.coordenadas);
    dinho(casa.coordenadas, casa);
}

```

```

    printf("%d_%d_%d\n", casa.numero, casa.coordenadas[X], casa.coordenadas[Y]);

    return 0;
}

void ta(Local casa) {
    casa.numero = 0;
    casa.coordenadas[X] = 100;
    casa.coordenadas[Y] = 200;
}

Local mu(Local y) {
    return (Local) {.numero = 155, .coordenadas[X] = 500, .coordenadas[Y] = 550};
}

void dan(int numero) {
    numero += 11;
}

int doo(int k) {
    return (k += 17);
}

void tu(int coordenadas[]) {
    coordenadas[X] *= 2;
    coordenadas[Y] /= 2;
}

void dinho(int coordenadas[], Local e) {
    coordenadas[X] -= 43;
    coordenadas[Y] -= 29;
    e.coordenadas[X] += 3;
    e.coordenadas[Y] += 7;
    e.numero += 63;
}

```

| |
|--|
| |
| |
| |

6. Analise o algoritmo abaixo e realize o teste de mesa a fim de determinar qual será a saída quando a entrada for os dígitos do seu Registro do Aluno (os seis dígitos). Você deve apresentar a impressão no quadro abaixo do código, no mesmo padrão no qual os valores são exibidos a partir do algoritmo.

```

#include <stdio.h>

#define MAX 6

typedef struct {

```

```

    int m[MAX];
} misterio;

int a, b, c;

void mis(misterio x);

misterio te(misterio y);

misterio rio(misterio z);

int main() {
    misterio s;
    for (a = 0; a < MAX; a++)
        scanf("%1d", &s.m[a]);
    mis(te(rio(s)));
    return 0;
}

void mis(misterio x) {
    for (a = 0; a < MAX; a++)
        printf("%d_", x.m[a]);
    printf("\n");
}

misterio te(misterio y) {
    do {
        for (a = 0, b = 0; a < MAX; a++)
            if (y.m[a] < y.m[b])
                b = a;
        y.m[b] *= 2;
    } while (y.m[b] < 10);
    return y;
}

misterio rio(misterio z) {
    for (a = 0; a < MAX; a++)
        z.m[a]++;
    return z;
}

```

| |
|--|
| |
| |
| |
| |
| |

7. Analise as três funções abaixo e/ou realize um teste de mesa a fim de determinar como será o vetor resultante após a chamada de cada uma das funções (separadamente). Considere que a entrada será um vetor de 12 posições, no qual as 6 primeiras serão os dígitos de seu RA e

as 6 últimos serão a data de seu aniversário (no formato dd/mm/aa). Você deve apresentar o vetor resultante de cada função no quadro abaixo do código, e explicar o que o algoritmo está fazendo.

| |
|---------------------------------|
| RA e Data de Aniversário: _____ |
|---------------------------------|

```
#include <stdio.h>

#define MAX 12

void skywalker(int vetor[], int tamanho) {
    int d[MAX];
    for (int a = 0, b = tamanho / 2, c = 0; a < tamanho / 2; a++, b++, c++) {
        d[c] = vetor[a];
        d[++c] = vetor[b];
    }
    for (int a = 0; a < tamanho; a++) vetor[a] = d[a];
}

void kenobi(int vetor[], int tamanho) {
    for (int i1 = 0; i1 < tamanho; i1++)
        for (int i2 = tamanho - 1; i2 > i1; i2--)
            if (vetor[i2] < vetor[i1]) {
                vetor[i1] += vetor[i2];
                vetor[i2] = vetor[i1] - vetor[i2];
                vetor[i1] = vetor[i1] - vetor[i2];
            }
}

void vader(int vetor[], int tamanho) {
    int i = 0, j = tamanho - 1, flag, aux;

    do {
        for (flag = 0; i <= j && !flag; i++)
            if (vetor[i] % 2 == 0)
                flag = 1;

        if (flag)
            for (flag = 0; j >= i && !flag; j--)
                if (vetor[j] % 2 == 1) {
                    aux = vetor[i - 1];
                    vetor[i - 1] = vetor[j];
                    vetor[j] = aux;
                    flag = 1;
                }
    } while (flag && i <= j);
}
```

| Função | Vetor resultante | O que a função faz? |
|------------|------------------|---------------------|
| skywalker: | | |
| kenobi: | | |
| vader: | | |

8. Analise o algoritmo abaixo e/ou realize um teste de mesa a fim de determinar qual será a saída quando a entrada do código for os dígitos do seu RA separados pelo caractere de espaço. Você deve apresentar a impressão no quadro abaixo do código, no mesmo padrão no qual os valores são exibidos a partir do algoritmo.

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

```
#define MAX 10
```

```
int main() {
    int matriz[MAX][MAX], soma[MAX][MAX], output[MAX][MAX];
    int vet[6];
    int mis = 5, te, ri, o, so;

    for (te = 6; te > 0; te--)
        scanf("%d", &vet[te - 1]);

    for (te = 0, o = 0; te < mis; te++)
        for (ri = mis - 1; ri >= 0; ri--, o++) {
            o %= 6;
            matriz[te][ri] = vet[o];
        }

    for (te = 0; te < mis; te++)
        for (ri = 0; ri < mis; ri++)
            soma[te][ri] = te + ri;

    for (te = mis - 1, o = 0; te >= 0; te--, o++)
        for (ri = mis - 1, so = 0; ri >= 0; ri--, so++)
            output[o][so] = matriz[te][ri] + soma[te][ri];

    for (o = mis - 1; o >= 0; o--) {
        for (so = 0; so < mis; so++)
            printf("%02d□", output[o][so]);
        printf("\n");
    }
    return 0;
}
```

| |
|--|
| |
| |
| |
| |
| |

9. Analise o algoritmo abaixo e/ou realize um teste de mesa a fim de determinar qual(ais) será(ão) a(s) saída(s) quando a entrada do código for os dígitos do seu Registro do Aluno. Você deve apresentar a impressão no quadro abaixo do código, no mesmo padrão no qual os valores são exibidos a partir do algoritmo.

```
#include <stdio.h>

#define X 6

typedef struct {
    int a[X], b;
} RA;

void f1(RA mist, int erio[]) {
    int i, a;
    for (i = 0, mist.b = 1; i < X; i++) {
        a = mist.a[X - i - 1] - i;
        if (a < 0) a *= -1;
        mist.a[i] *= a;
        erio[i] = mist.a[i] % 10;
        mist.b += mist.a[i];
    }
}

int f2(int mist[], int erio) {
    for (int i = 0, erio = 0; i < X; i++) {
        mist[i]++;
        erio += mist[i];
    }
    return erio + 2 * X;
}

int main() {
    RA ra;
    int i, mist[X];

    for (i = 0, ra.b = 0; i < X; i++) {
        scanf("%1d", &ra.a[i]);
        mist[i] = ra.a[i];
        ra.b += ra.a[i];
    }

    f1(ra, mist);

    printf("%d␣%d\n", f2(ra.a, ra.b), ra.b);

    for (i = 0; i < X; i++) printf("%d␣", ra.a[i]);
    printf("\n");
    for (i = 0; i < X; i++) printf("%d␣", mist[i]);
    printf("\n");
}
```

```

    return 0;
}

```

| |
|--|
| |
| |
| |
| |
| |

10. Analise o algoritmo abaixo e/ou realize um teste de mesa a fim de determinar qual será a saída quando a entrada do código for os dígitos do seu Registro do Aluno. Você deve apresentar a impressão no quadro abaixo do código, no mesmo padrão no qual os valores são exibidos a partir do algoritmo.

```

#include <stdio.h>

#define MAX 6

int ra[MAX];
int misterio[MAX];

void mis(int x);
int te(int x);
int ri(int x, int y, int z, int w);
void o(int x, int y);

int main() {
    int a;
    for (a = 0; a < MAX; a++) {
        scanf("%1d", &ra[a]);
        misterio[a] = ra[a];
    }
    for (a = 0; a < MAX; a++) mis(ra[a] - 5);
    for (a = 0; a < MAX; a++) printf("%d□", misterio[a]);
    printf("\n");
    return 0;
}

void mis(int x) {
    int a, b = te(ri(x, 0, -1, 1));
    if (!x) return;
    for (a = 0; b != 0; a = b, b = te(a + ri(x, 0, -1, 1))) o(a, x);
}

int te(int x) {
    return (x % MAX + MAX) % MAX;
}

int ri(int x, int y, int z, int w) {

```

```

    if (x > y) return z;
    return w;
}

void o(int x, int y) {
    int a = misterio[x], b = te(x + ri(y, 0, 1, -1));
    misterio[x] = misterio[b];
    misterio[b] = a;
}

```

| |
|--|
| |
| |
| |
| |
| |

11. Analise o algoritmo abaixo e/ou realize um teste de mesa a fim de determinar qual(ais) será(ão) a(s) saída(s) quando a entrada para a variável ra for o seu Registro de Aluno (os seis dígitos juntos). Você deve apresentar a impressão no quadro abaixo do código, no mesmo padrão no qual os valores são exibidos a partir do algoritmo.

```

#include <stdio.h>

#define X 6

void f1(int a[], int b) {
    int i;
    for (i = 0; i < X; i++, b /= 10) a[X - i - 1] = b % 10;
}

int f2(int a) {
    int i, b;
    if (!a) return 1;
    for (i = a - 1, b = a; i > 1; i--) b *= i;
    return b;
}

int f3(int a[]) {
    int t, i, j, k, l, m, n = 0;
    for (t = 3; t < X + 1; t++)
        for (i = 0; i <= X - t; i++) {
            for (k = i, l = i + t - 1, m = 1; l > k && m; k++, l--)
                if (a[k] != a[l]) m = 0;
            if (m) n++;
        }
    return n;
}

int f4(int a[]) {

```

```

    int b;
    b = a[0] - a[5];
    if (b < 0) b = -b;
    b %= 3;
    b = f2(b) + f3(a);
    return b % (X + 1);
}

int main() {
    int ra, mist[X], erio, i;

    scanf("%d", &ra);

    f1(mist, ra);
    erio = f4(mist);

    if (erio == 0)
        printf("zero\n");
    else {
        for (i = 0; i < erio; i++) printf("%d", mist[i]);
        printf("\n");
    }
    return 0;
}

```

| |
|--|
| |
| |
| |
| |
| |

12. Complete a função abaixo, e o que mais for necessário, nos espaços em branco (nem todos os espaços em branco deverão ser preenchidos). Sua função receberá como parâmetro uma matriz quadrada e sua dimensão, e deverá ordenar a matriz em ordem decrescente (o maior número ficará na posição `matriz[0][0]`, o segundo maior em `matriz[0][1]`, e assim por diante...).

Complete o código a seguir nos locais indicados, se necessário

```

typedef struct {
    int lin, col;
} -----;

void ordenaMatrizDec(int matriz[MAX][MAX], int tamanho) {
    int maior, linha, coluna, aux;
    Coordenadas pos_maior, pos_inicial;

    -----
    pos_inicial.col = 0;

```

```

while (pos_inicial.lin < _____) {
    _____
    pos_maior.lin = pos_inicial.lin;
    pos_maior.col = pos_inicial.col;

    for (linha = pos_inicial.lin; linha < tamanho; linha++) {
        for (coluna = 0 _____; coluna < tamanho; coluna++){
            if (linha == pos_inicial.lin _____)
                coluna = pos_inicial.col;

            if (maior < _____) {
                maior = matriz[linha][coluna];
                pos_inicial.col = coluna;
                _____
            }
        }
    }

    aux = matriz[pos_maior.lin][pos_maior.col];
    _____
    matriz[pos_inicial.lin][pos_inicial.col] = aux;

    pos_inicial.col++;
    if (_____ ) {
        pos_inicial.col = 0;
        pos_inicial.lin++;
    }
}
}

```

13. Implemente um algoritmo que dados $2 \leq n \leq 10^3$, n par, pontos (x, y) inteiros) que dois a dois determinam um segmento de reta (sempre sequenciais na entrada, ex.: $P_1 P_2 P_3 P_4$, P_1 e P_2 determinam um e P_3 e P_4 um outro) calcule o número máximo de retângulos que é possível formar movimentando e/ou rotacionando os segmentos de reta pelo plano, não é necessária validação para entrada.

Já existe um código previamente escrito, você deve entender a lógica de implementação utilizada e completar onde indicado para finalizar o algoritmo e deixá-lo funcional, resolvendo o problema proposto.

Complete o código a seguir nos locais indicados, se necessário

```

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

#define TAM 1000

typedef struct {
    int x;

```

```

    int y;
} -----;

typedef struct {
    Ponto p1;
    -----;
    ----- tamanho;
    int verificado;
} Segmento;

double calcTamanho(-----) {
    double tamanho;
    tamanho = -----
    -----
    return -----;
}

int obtemPares(-----, int n) {
    int i, j, iguais, pares;
    for (i = 0, -----; -----; i++)
        if (!segs[i].verificado) {
            -----
            for (j = i + 1, iguais = ----; j < n; j++) {
                if (segs[j].tamanho -----) {
                    segs[j].verificado = 1;
                    -----
                }
            }
            pares += iguais / 2;
        }
    return -----
}

int main() {
    Segmento -----[TAM / 2];
    int ----, i, j, iguais, pares, -----;

    scanf("%d", &n);
    -----

    for (i = 0; i < n; i++) {
        scanf("%d %d", -----, -----);
        scanf("%d %d", -----, -----);
        ----- = calcTamanho(segs[i]);
        segs[i].-----
    }

    total = obtemPares(segs, -----) / 2;

    printf("%02d\n", -----);
}

```



```

    return 0;
}

```

14. Complete a função abaixo, e o que mais for necessário, nos espaços em branco (nem todos os espaços em branco deverão ser preenchidos). Sua função receberá como parâmetros a dimensão de uma matriz e em seguida seus valores. Os valores da matriz representarão a altura de prédios. O programa deve imprimir as coordenadas dos prédios em ordem decrescente de altura.

Complete o código a seguir nos locais indicados, se necessário

```

#include <stdio.h>

#define MAX 100

typedef struct {
    int x, y;
} Coordenadas;

int tam_matriz;
float altura_predios[MAX][MAX];

Coordenadas criar_coordenada(int x, int y);

float altura_predio(Coordenadas coordenada);

void ler_matriz();

Coordenadas obter_maior_predio();

void obter_outros_predios(Coordenadas maior_predio);

int main() {
    ler_matriz();
    obter_outros_predios(obter_maior_predio());
    return 0;
}

Coordenadas criar_coordenada(int x, int y) {
    -----
    -----
    -----
    return -----
}

float altura_predio(Coordenadas coordenada) {
    -----
}

void ler_matriz() {
    scanf("%d", &tam_matriz);
}

```

```

    for ( _____ )
        for ( _____ )
            scanf("%f", &altura_predios[i][j]);
}

Coordenadas obter_maior_predio() {
    int i, j;
    Coordenadas maior_predio;
    _____

    for (i = 0; i < tam_matriz; i++) {
        for (j = 0; j < tam_matriz; j++) {
            if (altura_predios[i][j] _____)
                _____
        }
    }

    printf("Maior predio: (%d,%d).\n", _____,
        _____);
    return _____;
}

void obter_outros_predios(Coordenadas maior_predio) {
    int i, j;
    int ordem_predio = 2, existe_menor;
    Coordenadas predio_atual, prox_predio;
    _____

    do {
        existe_menor = 0;
        for (i = 0; i < tam_matriz; i++)
            for (j = 0; j < tam_matriz; j++)
                if (altura_predios[i][j] _____) {
                    if (existe_menor) {
                        if (altura_predios[i][j] _____)
                            prox_predio = _____;
                    } else
                        prox_predio = _____;
                    existe_menor = 1;
                }

        if (existe_menor) {
            printf("%do maior predio: (%d,%d).\n",
                _____, prox_predio.x, prox_predio.y);
            _____
            predio_atual = prox_predio;
        }
    } while (existe_menor);
}

```

15. Complete a função abaixo, e o que mais for necessário, nos espaços em branco (nem todos os espaços em branco deverão ser preenchidos). Seu programa deve receber duas Strings, uma por linha, e imprimir se elas são anagramas.

Complete o código a seguir nos locais indicados, se necessário

```

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

#define MAX 200

int  checarAnagrama(_____);

int  main() {
    char  str1[MAX], str2[MAX];

    scanf("_____", str1, str2);

    if (checarAnagrama(str1, str2))
        printf("As strings são anagramas.\n");
    else
        printf("As strings NÃO são anagramas.\n");

    return 0;
}

int  checarAnagrama(char str1[], char str2[]) {
    int  frequenciaChar1[256] = {0}, frequenciaChar2[256];
    _____
    _____

    if (strlen(str1) != _____) return 0;

    for (i = 0; str1[i] != '\0'; i++)
        _____

    for (i = 0; str2[i] != '\0'; i++)
        _____

    for (i = 0; i < _____; i++)
        if (frequenciaChar1[i] != frequenciaChar2[i])
            return 0;

    return 1;
}

```

16. Complete a função abaixo, e o que mais for necessário, nos espaços em branco (nem todos os espaços em branco deverão ser preenchidos). Seu programa deve copiar uma quantidade de bytes de uma origem de um arquivo para outro. Seu programa deve receber o nome de dois arquivos, a posição de origem (a partir de que byte será copiado) e a quantidade de bytes;

Complete o código a seguir nos locais indicados, se necessário

```

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

int main() {
    FILE *fp1;
    FILE *fp2;

    int count = 0, posicao = 0, qtdBytes = 0;
    char arquivo1[200], arquivo2[200], buffer[1024];

    scanf("%[^\\n]s%c%[^\\n]s", arquivo1, arquivo2);

    fp1 = fopen(______);
    if (!fp1) {
        printf("Arquivo\\ \"%s\" \\_nao\\_encontrado\\n", arquivo1);
        return -1;
    }

    fseek(fp1, 0, SEEK_END);

    _____
    count = _____(fp1);
    scanf("%d%d", &posicao, &qtdBytes);

    fp2 = _____
    if (!fp2) {
        printf("Arquivo\\ \"%s\" \\_nao\\_pode\\_ser\\_aberto\\n", arquivo2);
        return -1;
    }

    fseek(fp1, posicao, _____);

    _____

    fwrite(buffer, qtdBytes, 1, fp2);

    buffer[qtdBytes] = '\\0';

    printf("Copiado: \\_\"%s\" \\_\\n", buffer);

    _____
    _____

    return 0;
}

```

17. Analise o algoritmo abaixo e/ou realize um teste de mesa a fim de determinar qual(ais) será(ão) a(s) saída(s) quando a entrada para a variável ra for o seu Registro de Aluno (os seis dígitos juntos). Você deve apresentar a impressão no quadro abaixo do código, no mesmo padrão no qual os valores são exibidos a partir do algoritmo.

```

#include <stdio.h>

```

```

#include <math.h>

void naoSei(int num, int enLimit);
int seiLa(int n1);
int h(int x);
int w(int n);
int x(int c);
int y(int n1[], int a);
int z(int dno);

int main() {
    int ra[6], a = 5;

    for (int i = 0; i < 6; i++) scanf("%1d", &ra[i]);

    naoSei(ra[0], ra[4] * 10 + ra[5]);
    ra[3] = y(ra, a);
    ra[4] = a + h(ra[2]);
    for (i = 0; i < 6; i++) printf("%d\n", ra[i]);
    return 0;
}

void naoSei(int num, int num2) {
    while (num <= num2) {
        if (seiLa(num)) printf("%d□", num);
        num += 7;
    }
}

int seiLa(int n1) {
    int num = 0;

    for (int i = 1; i < n1; i++)
        if (n1 % i == 0) num += i;
    if (num == n1) return 1;
    else return 0;
}

int y(int n1[], int a) {
    n1[0] = z(n1[5]);
    n1[1] = w(n1[2]);
    a = x(n1[3]);
    return w(n1[5]);
}

int z(int dno) {
    int bno = 0, remainder, f = 1;
    while (dno != 0) {
        remainder = dno % 2;
        bno = bno + remainder * f;
        f = f * 10;
        dno = dno / 2;
    }
}

```

```

    }
    return bno;
}

int w(int n) {
    int num = 0, f = 1;
    while (num <= n - 1) {
        f = f + f * num;
        num++;
    }
    return f;
}

int x(int c) {
    int n = 0;
    for (int i = 1; i < c; i++) n += w(i) / i;
    return n;
}

int h(int x) {
    int sum = x, m = -1, ctr, mm, nn;
    for (int i = 1; i < 5; i++) {
        ctr = (2 * i + 1);
        mm = pow(x, ctr);
        nn = mm * m;
        sum = sum + nn;
        m = m * (-1);
    }
    return sum;
}

```

| |
|--|
| |
| |
| |
| |
| |
| |
| |

18. Analise o algoritmo abaixo e/ou realize um teste de mesa a fim de determinar qual(ais) será(ão) a(s) saída(s) quando a entrada para o vetor ra for o seu Registro de Aluno (os seis dígitos). Você deve apresentar a impressão no quadro abaixo do código, no mesmo padrão no qual os valores são exibidos a partir do algoritmo.

```

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

void faz_algo(char txt[][15], int ra[], int i);
void faz_uma_coisa_ae(char txt[][15], int i);

```


19. Complete o código abaixo, e o que mais for necessário, nos espaços em branco (nem todos os espaços em branco deverão ser preenchidos). Seu programa armazena uma matriz que representa bairros. Cada célula é um bairro que contém o orçamento atual, o ganho no mês e a perda no mês.

Após receber os valores da matriz, seu programa receberá o número de entradas e cada entrada pode ser 'g' para ganho de orçamento no bairro, 'p' para perda de orçamento no bairro e 't' para transferência de orçamento de um bairro a outro.

Para cada entrada do switch case, primeiro é recebido o valor de ganho, perda seguido do destino (linha e coluna da matriz) ou no caso da transferência é recebido o valor seguido da origem e destinos (linhas e colunas da matriz).

No fim, são impressos os valores de cada bairro (valor, ganhos e perdas) linha por linha.

Complete o código a seguir nos locais indicados, se necessário

```
#include <stdio.h>

typedef struct {
    ____ valor_atual;
    float ____;
    float perda_mes;
} Dados;

void ganho(____, float valor, int i, int j);
void perda(Dados bairros[][5], float valor, int i, int j);
void transferencia(Dados bairros[][5], float valor, int i,
                  int j, int m, int n);

int main() {
    ____ bairros[5][5];
    int entradas, m, n;
    float valor;
    char opc;

    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 5; j++){
            scanf("%f", &bairros[i][j].valor_atual);
            bairros[i][j].ganho_mes = 0;
            bairros[i][j].perda_mes = 0;
        }

    scanf("%d", &entradas);
    for (int i = 0; i < entradas; i++) {
        scanf("____", &opc);
        scanf("%f%d%d", &valor, &i, &j);
        switch (____) {
            case 'g':
                ____
                break;
        }
    }
}
```



```

        case 'p':
            -----
            break;

        case 't':
            scanf("%d%d", &m, &n);
            transferencia(bairros, valor, i, j, m, n);
            break;
    }
}

for (-----)
    for (-----) {
        printf("---Bairro_(%d,_%d)\n", i, j);
        printf("Valor:_.2f\n", bairros[i][j].-----);
        printf("Ganho:_.2f\n", bairros[i][j].-----);
        printf("Perda:_.2f\n", bairros[i][j].-----);
    }
}
-----

void ganho(Dados bairros[][5], float valor, int i, int j) {
    bairros[i][j].valor_atual += valor;
    bairros[i][j].ganho_mes += valor;
}

void perda(Dados bairros[][5], float valor, int i, int j) {
    -----
    -----
    -----
    -----
}

void -----(Dados bairros[][5], float valor, int i,
              int j, int m, int n -----) {
    bairros[m][n].valor_atual += valor;
    -----
    bairros[i][j].valor_atual -= valor;
    -----
}

```

20. Analise o algoritmo abaixo e/ou realize um teste de mesa a fim de determinar qual(ais) será(ão) a(s) saída(s) quando a entrada para o vetor ra for o seu Registro de Aluno (os seis dígitos juntos). Você deve apresentar a impressão no quadro abaixo do código, no mesmo padrão no qual os valores são exibidos a partir do algoritmo.

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

```

typedef struct {
    int a, b, c;
} Vet;

```

```

int somebody(int num) {
    return num * num % 10;
}

int once(int aha) {
    int result;

    result = (aha * 3) % 10;
    result += somebody(result);
    result %= 10;

    return result;
}

int told(Vet vet[]) {
    int L = vet[1].c, shape = L * L;

    vet[0].a += shape;
    vet[0].a %= 10;

    return L;
}

void me(Vet vet[]) {
    int x;

    vet[0].b = (vet[0].a + vet[1].a + 1) % 10;
    vet[0].a = once(vet[0].c + 3);
    vet[1].b = (vet[0].c + vet[1].c + 1) % 10;
    vet[1].a = once(vet[1].c + 3);
    vet[1].c = told(vet);
}

int main() {
    Vet ra[2];

    for (int i = 0; i < 2; i++) scanf("%1d%1d%1d", &ra[i].a, &ra[i].b, &ra[i].c);

    me(ra);

    for (int i = 0; i < 2; i++) printf("%d", ra[i].a, ra[i].b, ra[i].c);
    printf("\n");

    return 0;
}

```

| |
|--|
| |
| |
| |
| |
| |