Programação estruturada

(4.)

```
void cadastraCategoria (struct categoria v[], int *ata ){
 int codigo;

 IF (*ata >= TAM) {
  Printf("não é possível cadastrar, vetor cheio.\n);
  return;

 Printf("digite o codigo da nova categoria:");
 Scanf("%d", codigo);

 For (int i=0; i<*ata; i++){
    IF(V[i]. codigo == codigo ) {
     Printf("já existe uma categoria com esse codigo);
     Return; }}

V[*ata].codigo=codigo;
PrintF("digite o nome da categoria:);
ScanF("%. d\n, V[*ata]. nome);
(*ata)++;
PrintF("categoria cadastrada")
```

(2.)

```
void imprimirCategorias (struct v[], int ata ){
 IF (*ata==0) {
  Printf(:nenhuma categoria cadastrada);
 } else {
  Printf ("\n categorias:\n");
  For(int i=0; i<ata; i++) {
   Printf(" ~~int~~ codigo:%d | Nome:%s\n", v[i].codigo, v[i].nome);
  }}
 }
```

③ void imprimirProdutos (struct produto p[], int qtdProdutos,
struct categoria c[], int qtdCategorias) {

```
if (qtdProdutos == 0) {
    printf("Nenhum produto cadastrado.\n");
    return; }
printf("\n lista de produtos:\n");
for (int i=0; i < qtdProdutos; i++) {

  char nomeCategoria[] = "";
  for (int i = 0; i < qtdCategorias; i++) {
    if (c[i].codigo == p[i].categoria) {
    strcpy(nomeCategoria, c[i].nome);
    break; }}

  double convertePreco = p[i].preco/100.0;
  printf("\n codigo: %d", p[i].codigo);
  // ( ~~   %s , p[i].titulo );
  // ( ~~   %s , p[i].descricao);
  // ( ~~   %s , nomeCategoria);
  // ( ~~   %f , convertePreco);
```

④ void selectionSortPorDescricao (struct Produto v[], int qtd) {

```
  int i, j, minIndex;
  struct Produto p;

  for (i = 0; i < qtd - 1; i++) {
  minIndex = i;
  for (j = i+1; j < qtd; j++) {
    if (strcmp(v[j].descricao, v[minIndex].descricao) < 0) {
      minIndex = j;
  }}
```

```
IF(minIndex != i) {
    P = V[i];
    V[i] = V[minIndex];
    V[minIndex] = P; }

Printf("Produtos ordenados:\n", imprimirProdutos(Produtos, qtd));
Return 0; }

void BuscaBinariaPorDescricao (struct Produto V[], int qtd, char *x) {
int inicio = 0;
int fim = qtd - 1;
int meio;

While (inicio <= fim) {
    meio = (inicio + fim) / 2;

int comparacao = strcmp(x, V[meio].descricao);

IF (comparacao == 0) {
  Printf("Produto encontrado\n");

  (Chamar void imprimirProdutos da questão 3 )

} else {
  Printf("Produto não encontrado\n");
}
```