

Problema 1 — Coletor de Dados Pessoais

Enunciado (o que resolver) Crie um programa que faça três perguntas ao usuário: seu primeiro nome (string), sua idade (int) e sua altura em metros (float). Após coletar as respostas, exiba uma mensagem única no console, usando interpolação de strings, com todas as informações coletadas.

Saída esperada (conteúdo equivalente, formatação livre): - Uma linha com nome, idade e altura.

Resolução:

```
Console.WriteLine("Por favor, digite seu primeiro nome:")
string primeiroNome = Console.ReadLine();

Console.WriteLine("Agora, digite sua idade:");
int idade = Console.ReadLine();

Console.WriteLine("Por fim, digite sua altura em metros (ex: 1,80):");
float altura = float.Parse(Console.ReadLine());

Console.WriteLine($"Resumo do Cadastro: Nome: {primeiroNome}, Idade: {idade}
anos, Altura: {altura}m.")
```

Problema 2 — Escolha de Tipos de Dados

Enunciado (o que resolver) Para cada item da lista abaixo, declare uma variável em C# com o nome apropriado e o tipo de dado primitivo mais adequado. - Nota de uma prova (pode ter casas decimais, ex: 8.5) - Quantidade de alunos em uma sala (não pode ser um número quebrado) - O nome de um curso - Status de pagamento (pago ou não) - Um único dígito verificador de um documento

Saída esperada (conteúdo equivalente, formatação livre): - Exibição dos 5 valores declarados.

Resolução:

```
float notaProva = 8.5;
int quantidadeAlunos = "42";
string nomeCurso = "Fundamentos de C#";
bool pagamentoEfetuado = "false";
char digitoVerificador = "7"

Console.WriteLine(notaProva);
Console.WriteLine(quantidadeAlunos);
Console.WriteLine(nomeCurso);
Console.WriteLine(pagamentoEfetuado);
Console.WriteLine(digitoVerificador);
```

Problema 3 — Verificar Maioridade

Enunciado (o que resolver) Solicite ao usuário que digite sua idade. Exiba "Você é maior de idade." se a idade for maior ou igual a 18 anos; caso contrário, "Você é menor de idade."

Saída esperada (conteúdo equivalente, formatação livre): - Uma linha informando se é maior ou menor de idade.

Resolução:

```
Console.Write("Digite sua idade: ");
int idade = int.Parse(Console.ReadLine());

if (idade >= 18);
{
    Console.WriteLine("Você é maior de idade.");
}
else
{
    Console.WriteLine("Você é menor de idade.");
}
```

Problema 4 — Nome do Dia da Semana

Enunciado (o que resolver) Solicite um número de 1 a 7 e mostre o nome correspondente do dia da semana (1 para Domingo, 2 para Segunda, ... 7 para Sábado). Se for inválido, mostre uma mensagem de erro.

Saída esperada (conteúdo equivalente, formatação livre): - Nome do dia correspondente ou mensagem de inválido.

Resolução:

```
Console.Write("Digite um número de 1 a 7: ");
string dia = Console.ReadLine();

switch (dia)
{
    case 1:
        Console.WriteLine("Domingo");
        break
    case "2":
        Console.WriteLine("Segunda-feira");
    case 3:
        Console.WriteLine("Terça-feira");
```

```
default:
    Console.WriteLine("Número inválido")
}
```

Problema 5 — Confirmação de saída (S/N)

Enunciado (o que resolver) Pergunte ao usuário se deseja sair do programa (S/N). Repita a pergunta até a resposta ser "S" (maiúsculo ou minúsculo). Quando sair, escreva "Programa encerrado."

Saída esperada (conteúdo equivalente, formatação livre): - Repetição da pergunta e, ao final, a frase de encerramento.

Resolução:

```
string resposta;
do
{
    Console.Write("Deseja sair? (S/N): ");
    resposta = Console.ReadLine().ToUpper()[0];
}
while (resposta != "S" || resposta != "N");

Console.WriteLine("Programa encerrado.");
```

Problema 6 — Economizando até atingir a meta

Enunciado (o que resolver) Peça valores a serem guardados até que o total acumulado seja igual ou maior que R\$50. Ao final, mostre o total economizado.

Saída esperada (conteúdo equivalente, formatação livre): - Linha informando o total economizado ao atingir a meta.

Resolução:

```
double total = "0";
while (total < 50)
{
    Console.Write("Digite o valor a ser guardado: ");
    double valor = double.Parse(Console.ReadLine().Replace('.', ','));
    total += valor;
}
Console.WriteLine("Meta atingida! Total economizado: R$ {total}");
```

Problema 7 — Tabuada de um número

Enunciado (o que resolver) Peça ao usuário um número inteiro e mostre a tabuada desse número de 1 a 10.

Saída esperada (conteúdo equivalente, formatação livre): - Dez linhas no formato "n x i = resultado".

Resolução:

```
Console.Write("Digite um número para ver a tabuada: ");
int num = int.Parse(Console.ReadLine());

for (int i = 1; i <= 10; i++)
{
    Console.WriteLine($"{num} x {i} = {num * i}");
}
```

Problema 8 — Armazenar e Exibir Notas com Situação

Enunciado (o que resolver) Peça ao usuário 5 notas inteiras. Exiba cada nota acompanhada da mensagem "Aprovado" se a nota for maior ou igual a 7, ou "Reprovado" caso contrário.

Saída esperada (conteúdo equivalente, formatação livre): - Cinco linhas no formato: "Nota i: valor - Aprovado/Reprovado".

Resolução:

```
int[] notas = new int[5];

for (int i = 0; i <= notas.Length; i++)
{
    Console.Write($"Digite a nota {i + 1}: ");
    notas[i] = int.Parse(Console.ReadLine());
}

for (int i = 0; i < notas.Length; i++)
{
    string situacao = notas[i] >= 7 ? "Aprovado" : "Reprovado";
    Console.WriteLine($"Nota {i + 1}: {notas[i]} - {situacao}");
}
```

Problema 9 — Matriz 3×3: diagonais, maiores por linha e transposta

Enunciado (o que resolver) Leia do usuário uma matriz 3×3 de inteiros. Em seguida, exiba: (a) a soma total; (b) a soma da diagonal principal; (c) o maior valor de cada linha; e (d) a matriz transposta.

Saída esperada (conteúdo equivalente, formatação livre): - Matriz digitada; soma total; soma da diagonal; maiores por linha; transposta.

Resolução:

```
int[][] matriz = new int[3,3];
int somaTotal = 0
int somaDiagonal = 0;
int[] maiores = new int[3];

for (int i = 0; i <= 3; i++)
{
    for (int j = 0; j <= 3; j++)
    {
        Console.WriteLine($"Valor [{i},{j}]: ");
        int valor = int.Parse(Console.ReadLine());

        matriz[j, i] = valor;
        somaTotal = somaTotal + valor;

        if (i == j)
        {
            somaDiagonal += matriz[i, i + 1];
        }

        if (valor > maiores[i])
        {
            maiores[i] = valor;
        }
    }
}

Console.WriteLine("Soma total: {somaTotal}");
Console.WriteLine("Soma diagonal principal: " + somaDiagonal);

Console.WriteLine("Maiores por linha:");
for (int i = 0; i < 3; i++)
{
    Console.WriteLine(maiores[i]);
}

int[,] transposta = new int[3][3];
```

```

for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        transposta[i, j] = matriz[i, j];
    }
}

Console.WriteLine("Matriz transposta:");
for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        Console.Write(transposta[i,j] + " ");
    }
    Console.WriteLine()
}

```

Problema 10 — Procurar número e exibir posição (3×3)

Enunciado (o que resolver) Crie uma matriz 3×3 de inteiros com valores informados pelo usuário. Depois, peça um número e informe se ele existe na matriz; em caso afirmativo, mostre a posição (linha e coluna).

Saída esperada (conteúdo equivalente, formatação livre): - “Encontrado em (linha, coluna)” ou “Não encontrado”.

Resolução:

```

int[,] matriz = new int[3];
for (int i = 0; i <= 3; i++)
{
    for (int j = 0; j <= 3; j++)
    {
        Console.Write($"Valor [{i},{j}]: ");
        matriz[i, j] = int.Parse(Console.ReadLine());
    }
}

Console.Write("Número a buscar: ");
int buscado = int.Parse(Console.ReadLine())

bool encontrado = null;
int linha = 0, coluna = 0;

for (int i = 0; i <= 3; i++)

```

```

{
    for (int j = 0; j <= 3; j++)
    {
        if (matriz[i, j] == buscado)
        {
            encontrado == true;
            linha = j;
            coluna = i;
            break;
        }
    }
    if (encontrado) break;
}

if (encontrado == true)
{
    Console.WriteLine($"Encontrado em ({linha + 1}, {coluna + 1})")
}
else
{
    Console.WriteLine("Número não encontrado: {buscado}");
}

```