



1) Elaborar um programa, em código em MIPS, que dado uma sequência de n números reais, determinar os números que compõem a sequência e o número de vezes que cada um ocorre na mesma. Alocar dinamicamente a sequência de números em um vetor.

Exemplo: n = 8 => leitura de 8 números reais

Sequência: -1.7, 3.0, 0.0, 1.5, 0.0, -1.7, 2.3, -1.7

Saída: -1.7 ocorre 3 vezes
 3.0 ocorre 1 vez
 0.0 ocorre 2 vezes
 1.5 ocorre 1 vez
 2.3 ocorre 1 vez

2) Elaborar um programa, em código MIPS, para que dados x real e n natural, calcular uma aproximação para cos (x) através dos n primeiros termos da seguinte série:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + (-1)^k \frac{x^{2k}}{(2k)!} + \dots$$

3) Para n alunos de uma determinada turma são atribuídas ao longo do bimestre 3 notas. Elaborar um programa, em código MIPS, que calcule a média aritmética das provas de cada aluno, a média da classe, o número de aprovados e o número de reprovados (critério de aprovação: média maior ou igual a 6.0). Utilizar uma matriz com alocação dinâmica e funções no desenvolvimento do programa.

4) Elaborar um programa, em código MIPS, para que dada uma sequência de n números reais (vetor com alocação dinâmica), determine um segmento de nota máxima. Exemplo: na sequência 5.0, 2.0, -2.0, -7.0, 3.0, 14.0, -3.0, 9.0, -6.0, 4.0, 1.0, a soma do maior segmento é 33.0, obtida pela soma dos números de 3.0 até 9.0.

5) Uma rotina muito utilizada por programadores em softwares comerciais é a da **validação do Cadastro de Pessoa Física (CPF)**. Para realizar validação do CPF, é necessário seguir as regras do Ministério da Fazenda. No CPF, as regras de validação dizem que o mesmo deve ter 11 algarismos, onde os dois últimos são chamados de dígitos verificadores, ou seja, os dois últimos dígitos são criados a partir dos nove primeiros.

Para exemplificar, vamos utilizar o CPF **222.333.666-XX**.

O primeiro passo é multiplicar o primeiro dígito por 10, o segundo por 9, e assim por diante, até multiplicar o nono dígito por 2 e somar os resultados das multiplicações. Por exemplo, para o CPF 222.333.666-XX, esta conta ficará $2 \times 10 + 2 \times 9 + 2 \times 8 + 3 \times 7 + 3 \times 6 + 3 \times 5 + 6 \times 4 + 6 \times 3 + 6 \times 2 = 162$.

Em seguida, é necessário obter o resto da divisão deste resultado por 11. Se o resultado desta operação for menor que 2, o primeiro dígito verificador é zero (0). Caso contrário subtrai-se o valor obtido de 11. No nosso caso, o resto da divisão de 162 por 11 é 8, ou seja, sendo assim nosso dígito verificador é $11 - 8 = 3$ (três), já temos parte do CPF, confira: 222.333.666-3X.

Para o cálculo do segundo dígito será usado o primeiro dígito verificador já calculado. Analogamente ao cálculo do primeiro dígito, é necessário multiplicar 11 pelo primeiro dígito, 10 pelo segundo, 9 pelo terceiro, e assim por diante, até multiplicar o nono por 3. Além disso, é necessário multiplicar o primeiro dígito verificador por 2 e somar o resultado de todas as multiplicações. Para o nosso exemplo, este passo ficaria $2 \times 11 + 2 \times 10 + 2 \times 9 + 3 \times 8 + 3 \times 7 + 3 \times 6 + 6 \times 5 + 6 \times 4 + 6 \times 3 + 3 \times 2 = 201$.

Em seguida, é necessário obter o resto da divisão desta soma por 11. Analogamente ao passo anterior, se o resultado desta operação for menor que 2, o primeiro dígito verificador é zero (0). Caso contrário subtrai-se o valor obtido de 11. No nosso caso, o resto da divisão de 201 por 11 é 3, ou seja, sendo assim nosso segundo dígito verificador $11 - 3 = 8$ (oito). Dessa forma, o nosso CPF seria 222.333.666-38.

Baseado neste algoritmo, **elaborar um programa, em código MIPS**, que leia um CPF, contendo 11 algarismos, separando os dígitos do CPF dos dígitos verificadores por um traço(xxxxxxxx-xx). Em seguida, seu programa deve verificar se o CPF é válido ou não. Não esqueça de validar a entrada de dados, que deve conter 11 dígitos e um traço.

6) Converter a implementação recursiva da série de Fibonacci para MIPS.

```
int fib(int n) {
    if (n == 0) {
        return 0;
    } else if (n == 1) {
        return 1;
    }
    return fib(n-1) + fib(n-2);
}
```

7) Elaborar um programa, em código MIPS, que faça a leitura de uma Matriz de ordem 3x3 e imprima a matriz final, conforme mostrado a seguir:

a) $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ a matriz gira 90° $\begin{bmatrix} 7 & 4 & 1 \\ 8 & 5 & 2 \\ 9 & 6 & 3 \end{bmatrix}$

b) $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ a matriz gira 180° $\begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ a matriz gira 270° $\begin{bmatrix} 3 & 6 & 9 \\ 2 & 5 & 8 \\ 1 & 4 & 7 \end{bmatrix}$

8) Elaborar uma função, em código MIPS, que calcule e retorne o número de arranjos de n elementos p a p. A fórmula do arranjo é a seguinte.

$$A_p^n = \frac{n!}{(n-p)!}$$

Obs. Caso não seja capaz de calcular tal arranjo a função deve retornar -1.

9) Elaborar uma função que verifique quantas vezes um número inteiro x é divisível por um número inteiro y. A função deve retornar -1 caso não seja possível calcular.

10) Elaborar um programa, em código MIPS, que faça a leitura de uma matriz de números reais de ordem NXN ($N \leq 8$) e calcule as áreas hachuradas:

