



UNIVERSIDADE DE COIMBRA  
FACULDADE DE CIÊNCIAS E TECNOLOGIA

**Departamento de Engenharia  
Informática**

## **Trabalho Prático #1 Algoritmos e Estruturas de Dados**

**2020-2021 – 2º Semestre**

**Submissão no Mooskak:**  
até 15mn após o final da aula

*Nota Importante: A fraude denota uma grave falta de ética e constitui um comportamento não admissível num estudante do ensino superior e futuro profissional licenciado. Qualquer tentativa de fraude pode levar a anulação da componente prática tanto do facilitador como do prevaricador.*

*A utilização de código disponível na net, desde que não infrinja direitos de autor é permitido neste trabalho. É assumido que este código foi compreendido com nível de detalhe idêntico ao esperado se tivesse sido desenvolvido pelo aluno. A explicação do código é matéria de interação com o docente da disciplina.*

### **Objectivos:**

Introdução ao Mooshak.

Tarefas:

- Soma de dois números
- Rotação de uma matriz

---

Os alunos deverão consultar previamente o texto de apoio sobre o Mooshak que acompanha este enunciado, e proceder ao seu registo no servidor Mooshak de AED:  
<https://mooshak.dei.uc.pt/~aed>.

Os problemas aqui apresentados servem de introdução à ferramenta Mooshak.

---

## Tarefa A :: Soma de dois números

Para esta tarefa o programa desenvolvido deve ler dois números do *stdin* e imprimir para o *stdout* a soma desses dois números. Esta tarefa tem como objectivo a familiarização com o mecanismo de input/output utilizado pelo *Mooshak*. Deste modo é disponibilizado no final do exercício código para a leitura e escrita do input e output, sendo que apenas terão de resolver o que está em falta. Mas antes disso a descrição do input e output é feita.

### Input

O *input* consiste numa linha com dois números inteiros com um espaço entre eles. Os números estão contidos entre 0 e 1000.

### Output

O *output* é dado por um número inteiro que resulta da soma dos dois números dados no input, seguido de um “\n” (linebreak).

### Exemplo

#### Input

2 4

#### Output

6

### Código de Suporte

#### C

```
#include <stdio.h>

int main() {
    int n1, n2;
    scanf("%d %d", &n1, &n2);

    int sum = .. // TODO

    printf("%d\n", sum);

    return 0;
}
```

#### C++

```
#include <iostream>

int main() {
    // Para uma leitura e escrita mais eficientes
    std::ios_base::sync_with_stdio(0);
    std::cin.tie(0);

    int n1, n2;
    std::cin >> n1 >> n2;

    int sum = .. // TODO

    std::cout << sum << "\n";

    return 0;
}
```

### Java (com uma classe para ler o input de forma mais eficiente)

**Nota 1:** O código deve estar contido numa só classe com o mesmo nome do ficheiro, neste caso Main.java

**Nota 2:** Programas em Java **não** devem incluir uma linha *package* no início do programa.

```
import java.util.*;
import java.io.*;

class Main {
    public static void main(String[] args) {
        InputReader in = new InputReader(System.in);
        PrintWriter out = new PrintWriter(System.out);

        int n1 = in.nextInt();
        int n2 = in.nextInt();

        int sum = .. // TODO

        out.println(sum);
        out.close();
    }

    static class InputReader {
        public BufferedReader reader;
        public StringTokenizer tokenizer;

        public InputReader(InputStream stream) {
            reader = new BufferedReader(new InputStreamReader(stream));
            tokenizer = null;
        }

        public String next() {
            while (tokenizer == null || !tokenizer.hasMoreTokens()) {
                try {
                    tokenizer = new StringTokenizer(reader.readLine());
                } catch (IOException e) {
                    throw new RuntimeException(e);
                }
            }
            return tokenizer.nextToken();
        }

        public int nextInt() {
            return Integer.parseInt(next());
        }
    }
}
```

### Python 3 (com métodos para leitura e escrita mais eficientes)

```
from sys import stdin, stdout

def readln():
    return stdin.readline().rstrip()

def outln(n):
    stdout.write(str(n))
    stdout.write("\n")

n1,n2 = [int(i) for i in readln().split()]
result = .. # TODO
outln(result)
```

### Tarefa B :: Rotação de uma matriz

O objectivo da tarefa B é construir um algoritmo que para uma dada matriz de inteiros proceda à sua rotação em 90º, 180º e 270º no sentido do ponteiro dos relógios e imprima as matrizes rodadas.

#### Input

O *input* começa com uma primeira linha que contém dois números inteiros que indicam o número de linhas e colunas da matriz. O tamanho máximo da matriz é 10x10. As restantes linhas descrevem a matriz, sendo que as colunas são espaçadas por um espaço e as linhas por “\n”.

#### Output

O *output* começa com o número “90” numa linha seguido da matriz rodada 90 graus. Depois o número “180” e a matriz rodada 180 graus. Por fim, temos o número “270” e a matriz rodada 270 graus. Nota: **não** devem existir espaços no final de cada linha.

#### Exemplos

##### Input

```
2 3
1 2 3
4 5 6
```

##### Output

```
90
4 1
5 2
6 3
180
6 5 4
3 2 1
270
3 6
2 5
1 4
```