

## ACH 2002 – Introdução à Análise de Algoritmos

EACH – SEGUNDO SEMESTRE DE 2025

Primeiro Exercício-Programa

Data de entrega: 12 de outubro de 2025.

Professor: Marcos Lordello Chaim

### Removendo Ilhas de Lixo

Autor: Lucas Leão Ferreira Barbosa

#### Introdução

O problema do lixo nos oceanos é um desastre que vem se agravando há décadas, e precisamos da sua ajuda para resolvê-lo! Para auxiliar na limpeza, drones especializados escanearam vastas áreas dos oceanos e enviaram dados sobre “ilhas de lixo”. Esses dados chegam em formato de matrizes, onde o número em cada célula representa a altura do lixo acumulado naquele metro quadrado ( $m^2$ ) específico.

O seu trabalho será desenvolver um programa que analise os dados de uma região escaneada para:

1. Identificar quantas ilhas de lixo distintas existem na região.
2. Calcular as dimensões de cada ilha para transporte em contêineres.
3. Determinar uma combinação de ilhas que possa preencher um contêiner exatamente.

Na matriz:

- O valor **0** indica mar (nenhum lixo).
- Qualquer valor maior que 0 indica lixo acumulado.

Duas células fazem parte da mesma ilha se estiverem conectadas horizontalmente ou verticalmente (não em diagonal).

#### O Exercício de Programação

##### Transporte e Cálculo

Os contêineres usados no transporte possuem **altura = 2 metros** e **largura = 3 metros**, mas a **profundidade é variável**. Como as ilhas de lixo são massas indivisíveis, cada ilha deve ser transportada inteiramente em um único contêiner.

Para calcular a profundidade necessária de um contêiner para armazenar uma ilha, utilizamos a seguinte fórmula:

$$\text{Profundidade} = \frac{(\text{soma das alturas das células da ilha})}{6}$$

O denominador 6 vem do fato de que a base do contêiner é sempre  $2 \times 3 = 6 \text{ m}^2$ , oriundo de um processo de compactação para encaixar as ilhas no molde do container. Assim, a soma das alturas é distribuída uniformemente pela área da base.

### Exemplo ilustrativo do cálculo

Se a ilha for formada pelas células  $\{1, 7\}$ :

$$\text{Profundidade} = \frac{1 + 7}{6} = \frac{8}{6} \approx 1$$

Se a ilha for formada pelas células  $\{4, 5, 6, 2, 4\}$ :

$$\text{Profundidade} = \frac{4 + 5 + 6 + 2 + 4}{6} = \frac{21}{6} = 3$$

### Observação sobre arredondamento:

- Se o valor da profundidade for decimal, considere apenas a parte inteira (conversão para `int`).
- Caso o valor calculado seja  $0 < r < 1$ , arredonde para cima (nesse caso,  $r = 1$ ).

### Entrada

O programa deverá ler os seguintes parâmetros:

1. Profundidade total do contêiner (inteiro).
2. Número de linhas da matriz.
3. Número de colunas da matriz.
4. Nome de um arquivo de texto (`.txt`) contendo a matriz (valores separados por espaço).

### Exemplo de arquivo de entrada:

```
1 7 0 0
0 0 2 4
4 5 6 0
```

Neste caso, existem duas ilhas:

- Ilha 1: 1, 7
- Ilha 2: 4, 5, 6, 2, 4

As profundidades calculadas serão:

- Ilha 1:  $(1+7)/6 = 8/6 = 1$
- Ilha 2:  $(4+5+6+2+4)/6 = 21/6 = 3$

## Saída

O programa deverá imprimir no terminal as seguintes informações:

1. **Linha 1:** O número total de ilhas encontradas.
2. **Linha 2:** As profundidades correspondentes a cada ilha, separadas por espaço (a ordem não importa).
3. **Linha 3:** Uma combinação de profundidades que some exatamente a profundidade do contêiner.  
**Se não houver combinação válida, imprimir: Nao ha resposta valida!**

## Execução e Avaliação

Seu programa deve ser feito apenas em linguagem C, compilado e executado via terminal. A nota será dividida igualmente entre os três requisitos da saída, e a correção será feita com diversos testes automatizados.

```
$ gcc <Numero_USP>.c -o EP1
$ ./EP1 <profundidade_container> <linhas_da_matriz> <colunas_da_matriz> exemplo.txt
```

## Exemplos de execução

As saídas do seu EP devem ser idênticas às descritas abaixo pois o EP será corrigido automaticamente. Logo, se a saída for diferente, sua nota poderá ser prejudicada. Observe que os caracteres da saída não utilizam acento. Nos exemplos a abaixo, suponha que o seu EP compilado se chame **EP1**.

Os exemplos abaixo mostram casos de teste, incluindo entrada, análise e saída.

### Exemplo 1

**Entrada:**

```
./EP1 4 3 4 arquivo.txt
arquivo.txt:
2 3 0 1
1 0 0 2
0 0 12 0
```

**Saída:**

```
3
1 1 2
2 1 1
```

## Exemplo 2

### Entrada:

```
./EP1 10 3 3 arquivo.txt
arquivo.txt:
5 5 0
0 4 0
0 0 9
```

### Saída:

```
2
2 1
Nao ha resposta valida!
```

## Exemplo 3

### Entrada:

```
./EP1 25 8 8 arquivo.txt
arquivo.txt:
13 4 0 0 36 0 1 87
0 4 3 0 0 23 0 0
0 0 9 0 0 0 0 0
0 0 0 0 0 0 0 0
2 4 5 0 0 34 21 8
4 3 0 0 12 0 8 1
0 9 0 0 0 0 0 2
0 0 0 12 0 0 0 0
```

### Saída:

```
7
5 9 14 4 2 12 2
14 9 2
```

## Exemplo 4

### Entrada:

```
./EP1 17 6 7 arquivo.txt
arquivo.txt:
0 30 12 0 0 0 0
6 24 0 0 42 48 0
0 0 0 0 54 0 0
0 18 0 0 0 0 24
```

```
0  0  0  12  12  0  0
36  0  0  0  0  0  6
```

### Saída:

```
7
12 24 3 4 4 6 1
12 4 1
```

## Observações importantes

### Sobre a avaliação

- O EP é individual. Não serão toleradas cópias! Exercícios copiados (com ou sem eventuais disfarces) receberão nota ZERO (inclusive o exercício original que foi copiado).
- Não será permitido a utilização de ferramentas de Inteligência Artificial para resolver o EP. Se for detectada a utilização dessas ferramentas, a nota será ZERO.
- Exercícios com erros de sintaxe (ou seja, erros de compilação) receberão nota ZERO.

### Sobre a entrega:

- O prazo de entrega é o dia 12/10/2025 às 23h59min.
- Vocês deverão fazer o “upload” do arquivo implementado por vocês com o seguinte nome <seu número USP>.c. Se o seu número USP é 1234567 então o seu arquivo deve ser nomeado 1234567.c. Os trabalhos serão corrigidos automaticamente; por isso, é essencial que vocês subam seus arquivos com o nome certo.

### Dicas:

A implementação do EP requer a manipulação de arquivos na linguagem C. Vocês podem utilizar a fonte abaixo:

- Arquivos: <https://linguagemc.com.br/arquivos-em-c-categoria-usando-arquivos/>

**Bom trabalho!**