



# OBI

## Aula 01

GEMP  
Grupo de Estudos da  
Maratona de Programação

- Três Fases
- Individual



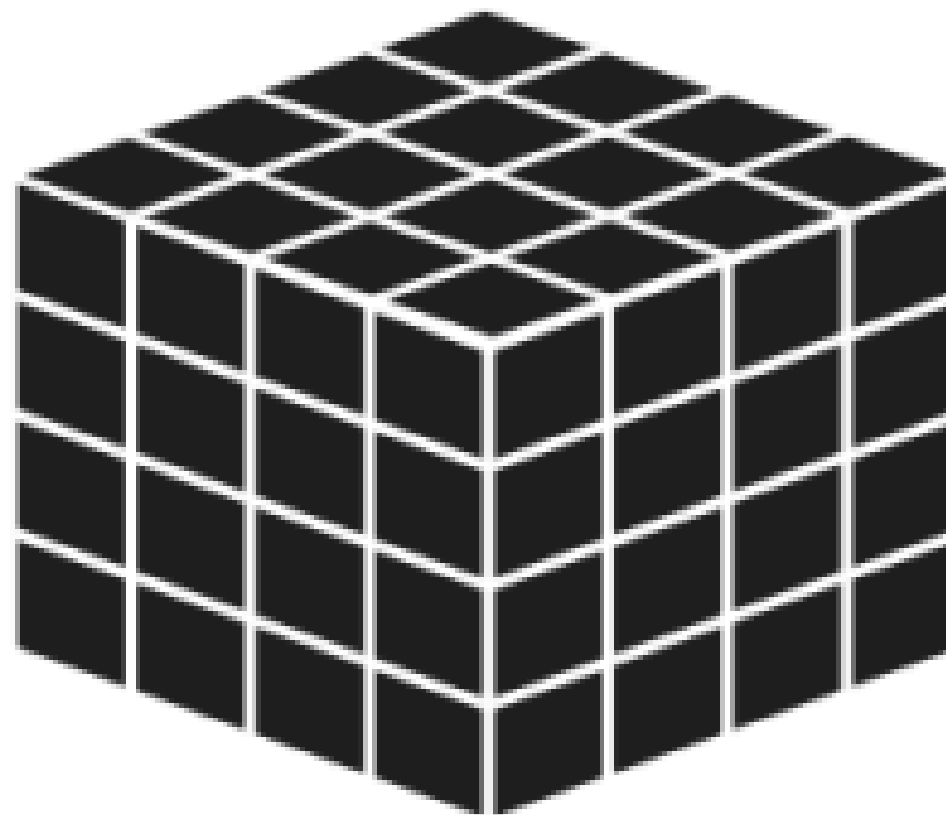
- Gratuita
- Duração 3h

# Conceitos de Matemática Discreta

- [P] Funções e relações
- [P] Ordem lexicográfica
- [P] Conjuntos (união/interseção, complemento)
- [P] Recursão (definições matemáticas recursivas)
- [P] Argumentos de contagem (princípio aditivo, princípio multiplicativo)
- [P] Permutações e função fatorial
- [P] Progressão aritmética
- [P] Princípio da casa dos pombos
- [P] Teoria dos jogos básica (posições vencedoras e perdedoras)
- [P1] Progressão geométrica
- [P1] Combinações
- [P1] Triângulo de Pascal e coeficientes binomiais
- [P1] Princípio da inclusão-exclusão

Ana comprou um cubo de madeira de lado  $N$  cm (ou seja, dimensões  $N \times N \times N$  centímetros) e o pintou todo de preto. Depois disso, ela cortou o cubo em  $N^3$  cubinhos de lado 1 cm (ou seja, dimensões  $1 \times 1 \times 1$  centímetro). Após o corte, alguns cubinhos terão nenhuma face pintada de preto, alguns terão exatamente uma face pintada, alguns terão exatamente duas faces pintadas e outros terão exatamente três faces pintadas.

Abaixo podemos ver um cubo de lado 4 cm ( $N = 4$ ) após Ana pintá-lo e cortá-lo.



**OBI 2024**

Ana contou quantas faces estavam pintadas em cada cubinho cortado do cubo acima e concluiu que, entre os 64 cubinhos, existem 8 cubinhos com nenhuma face pintada de preto, 24 cubinhos com exatamente uma face pintada, 24 cubinhos com exatamente duas faces pintadas e 8 cubinhos com exatamente três faces pintadas.



**Neps Academy**  
*(recomendo)*



**Beecrowd**

# Tipos de dados


Primitive type	Size	Minimum	Maximum	Wrapper type
<b>boolean</b>	—	—	—	<b>Boolean</b>
<b>char</b>	16 bits	Unicode 0	Unicode $2^{16}-1$	<b>Character</b>
<b>byte</b>	8 bits	-128	+127	<b>Byte</b>
<b>short</b>	16 bits	$-2^{15}$	$+2^{15}-1$	<b>Short</b>
<b>int</b>	32 bits	$-2^{31}$	$+2^{31}-1$	<b>Integer</b>
<b>long</b>	64 bits	$-2^{63}$	$+2^{63}-1$	<b>Long</b>
<b>float</b>	32 bits	IEEE754	IEEE754	<b>Float</b>
<b>double</b>	64 bits	IEEE754	IEEE754	<b>Double</b>
<b>void</b>	—	—	—	<b>Void</b>

FAST

RECM.





 hello\_world.c

```
#include <stdio.h>
```

```
int main () {
```

```
    printf("Hello World!\n");
```



```
    return 0;
```

```
}
```



CodeImage





C++ hello\_world.cpp

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    cout << "Hello World!\n";
```

```
    return 0;
```

```
}
```



CodeImage





hello\_world.py

```
print("Hello World!")
```



CodeImage



index.cpp

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int x = 1;
7      if (x > 0) {
8          cout << "positive\n";
9      }
10     else if (x < 0) {
11         cout << "negative\n";
12     }
13     else {
14         cout << "nulo\n";
15     }
16     return 0;
17 }
```



Codelmage



 index.cpp

```
1  #include <iostream>
2
3  int main() {
4      int x = 1;
5      if (x > 0) {
6          std::cout << "positive\n";
7      }
8      else if (x < 0) {
9          std::cout << "negative\n";
10     }
11     else {
12         std::cout << "nulo\n";
13     }
14     return 0;
15 }
```



 index.cpp

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int a, b;
7      cin >> a >> b;
8
9      if (a == 0 && b == 0) {
10         cout << "os dois sao zeros\n";
11     }
12     else if(a == 0 || b == 0) {
13         cout << "pelo menos um é igual a zero\n";
14     }
15     else if (!a && !b) {
16         cout << "o mesmo caso do primeiro\n";
17     }
18     return 0;
19 }
```



Codelmage

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      if (1) {
7          cout << "isso funciona\n";
8      }
9      if (0) {
10         cout << "isso não funciona\n";
11     }
12     return 0;
13 }
```

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      bool a;
7
8      if (a) {
9          cout << "isso funciona\n";
10     }
11     if (b) { *a em vez de b
12         cout << "isso não funciona\n";
13     }
14     return 0;
15 }
```

```
1  #include <iostream>
2
3  int main() {
4      int a, b;
5      std::cin >> a >> b;
6
7      if (a == 0 && b == 0) {
8          std::cout << "os dois sao zeros\n";
9      }
10     else if(a == 0 || b == 0) {
11         std::cout << "pelo menos um é igual a zero\n";
12     }
13     else if (!a && !b) {
14         std::cout << "o mesmo caso do primeiro\n";
15     }
16     return 0;
17 }
```



```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int a, b;
7      cin >> a >> b;
8
9      if (a % 2 == 0) {
10         cout << a << " é par\n";
11     }
12     else {
13         cout << a << " é ímpar\n";
14     }
15     if (a % b == 0) {
16         cout << a << " é divisível por " << b << "\n";
17         cout << b << " é múltiplo de " << a << "\n";
18     }
19     if (!(a%b)) {
20         cout << a << " é divisível por " << b << "\n";
21         cout << b << " é múltiplo de " << a << "\n";
22     }
23     return 0;
24 }
```

# Primeiros problemas

Neps

- **Positivo, Negativo ou nulo**
- **Móbile**
- **Acelerador de Partículas**
- **Ordenação de Três Números**