

Introdução

#estrutura_dados_competicoes #faculdade

- Plano de Ensino
- Repositório: <https://github.com/edsomjr/TEP>
- Linguagens: **C, C++, Python**

Introdução à Programação Competitiva

Programação competitiva: resolver problemas conhecidos da Ciência da Computação, o mais rápido possível

- Desenho de algoritmos (resolução de problemas e pensamento matemático)
- Implementação de algoritmos (habilidades de programação)

ACM - <https://www.acm.org/>

ICHC - <https://icpc.global/>

Maratona de Programação da SBC - <https://maratona.sbc.org.br/>

Juízes eletrônicos

Correção automática para problemas de programação competitiva.

- Beecrowd - <https://judge.beecrowd.com/pt/login>
- Codeforces - <https://codeforces.com/>
- AtCoder - <https://atcoder.jp/>

AtCoder Beginner Contest

Formatos de entrada/saída

Categoria 1:

```
int main() {
    int x, y;
    scanf("%d %d", &x, &y)
    printf("%d", x + y)
    return 0;
}
```

Categoria 2:

```

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

    while (T--) {
        int X, Y;
        scanf("%d %d", &X, &Y);
        printf("%d", X + Y);
    }

    return 0;
}

```

Categoria 3:

```

int main() {
    int X, Y;

    while (scanf("%d %d", &X, &Y), X != -1 && Y != -1) {
        print("%d\n", X + Y);
    }

    return 0;
}

```

Categoria 4:

```

int main() {
    int X, Y;

    while (scanf("%d %d", &X, &Y) == 2) {
        printf("%d\n", X + Y);
    }
}

```

Em C++ é necessário a seguinte linha para evitar problemas de performance com a entrada padrão:

```
ios::sync_with_stdio(false);
```

O C++ usa `cin` e `cout`, normalmente a linguagem tenta sincronizar os dois com o `stdin` e `stdout` porém isso gera problemas de performance, portanto é necessário desativar a sincronização.

Em competições também é evitado usar o `endl`

Assinatura do operador `>>` :

```
istream& operator >>(istream& is, const int &x);
```