

EXERCÍCIOS DE NÍVEL DE PROVA

Lembrando que o objetivo da disciplina é:

- Ser capaz de resolver exercícios de programação (de nível básico ou intermediário) eventualmente que não sigam nenhum exemplo previamente visto na disciplina

Em outras palavras, não há "receitas", "macetes", "esquemas prévios"

Isso não implica que os problemas sejam trabalhosos, extensos, ou sofisticados

O princípio é que o aluno possa mobilizar as estruturas básicas de qualquer linguagem de programação em um tipo de problema novo (desconhecido)

Então, surgem os Exercícios de Nível de Prova

- Problemas não necessariamente sofisticados, mas diferentes, que exigem a articulação das estruturas básicas de programação

COMO RESOLVER UM ENP?

Os ENP surgem também para incentivar uma atitude fundamental proposta nestes ATP:

- Um problema pode levar mais que 10 min

Em outras palavras:

- Cada aluno, ciente de suas possibilidades, recursos e limitações, deve possuir uma série de exercícios em aberto (a "malinha invisível de desafios")
- Quanto mais o aluno conseguir resolver esses exercícios com o próprio esforço ("andar com as próprias pernas"), melhor
 - "Andar com as próprias pernas" significa evitar, ao máximo, ver a resposta pronta de um problema, seja "no fim do livro", na internet, ou a resposta de um colega
 - Essas respostas não são proibidas, mas se não há uma mescla entre ver respostas e andar com as próprias pernas, isso é um mau sinal
 - Em vendo uma resposta, buscar um exercício equivalente para, desta vez, resolver por conta própria
- Incentivar "a paciência da resposta", a "construção de uma saída"
- Como se fosse um quebra-cabeça, um passatempo



Dessa forma, os ENF não têm prazo de entrega

- Nem acumule ENF demais: tenha uma quantidade de ENF ao seu alcance

EXEMPLO: DIVISÃO INTEIRA300) Programa divide

Faça um programa que, a partir de dois números inteiros positivos fornecidos, mostra o quociente e o resto da divisão inteira do primeiro pelo segundo. As operações de multiplicação e divisão disponíveis na linguagem de programação não podem ser empregadas na resposta. Um algoritmo básico para realizar a divisão inteira consiste em subtrair o divisor do dividendo quantas vezes for possível.

Exemplos de uso:

```
$ ./divide
Informe os numeros inteiros positivos: 20 6
Q = 3 R = 2

$ ./divide
Informe os numeros inteiros positivos: 683 47
Q = 14 R = 25

$ ./divide
Informe os numeros inteiros positivos: 0 3
Dados de entrada invalidos
```

Apresentação informal do algoritmo

```
20
-6 =          1 vez
14 (< 0? Não)
-6 =          2 vezes
8 (< 0? Não)
-6 =          3 vezes
2 (< 0? Não)
-6 =          4 vezes
-4 (< 0? Sim)
```

```
Q = 4 - 1 = 3
R = -4 + 6 = 2
```

Resposta

divide.c

```
// Gabarito oficial
#include <stdio.h>

int main() {
    int dividendo, divisor;
    printf("Informe os numeros inteiros positivos: ");
    scanf("%d %d", &dividendo, &divisor);
    if (dividendo <= 0 || divisor <= 0) {
        printf("Dados de entrada invalidos\n");
        return 1;
    }

    int a = dividendo, b = 0;
    while(a >= 0) {
        a = a - divisor;
        b = b + 1;
    }

    printf("Q = %d R = %d\n", b - 1, a + divisor);
}
```