

### **MCTA028 – Programação Estruturada**

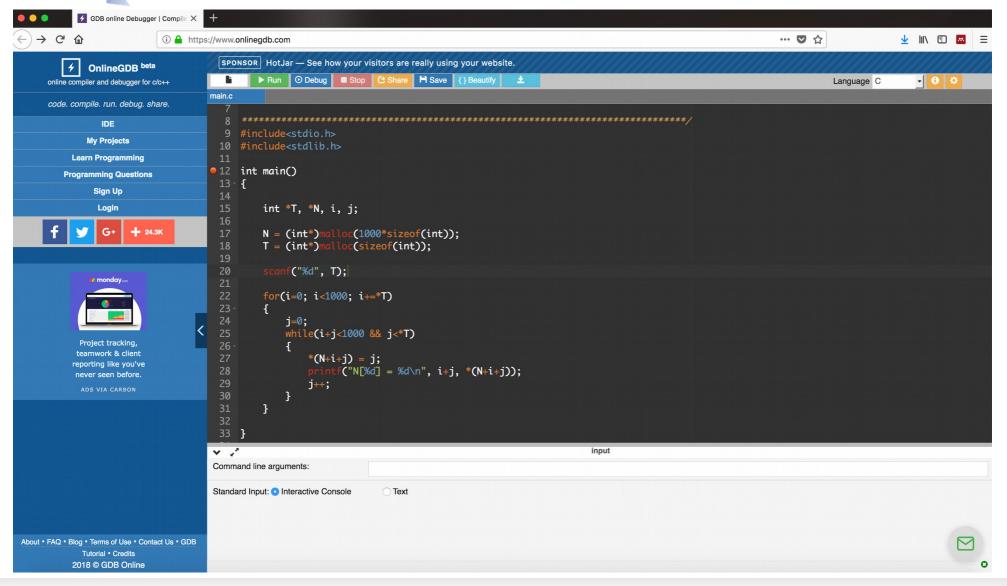
# Aula 05: Laboratório - Recursão e ponteiros (parte 2)

Prof. Francisco Fraga e Prof. Mario Gazziro francisco.fraga@ufabc.edu.br mario.gazziro@ufabc.edu.br 3Q-2018

Slides adaptados dos originais gentilmente fornecidos pelo Prof. Jesús P. Mena-Chalco



https://www.onlinegdb.com/



## Lista de exercícios para entregar

Nesta lista serão trabalhados os tópicos Recursão e Ponteiros

Usaremos a Plataforma URI para a avaliação de todos os problemas da lista: https://www.urionlinejudge.com.br.

- 1. Problema 1059. Números pares. Crie uma função recursiva para resolver este problema.
- 2. Problema 1073. Use ponteiro pelo menos uma vez no código.
- 3. Problema 1074. Par ou ímpar. Use ponteiro pelo menos uma vez no código.
- 4. Problema 1157. Divisores I. Escreva uma função recursiva para resolver este problema.
- 5. Problema 1164. Divisor perfeito. Use ponteiro pelo menos uma vez no código. Sugestão: use a função recursiva criada no Problema 1157.
- 6. Problema 1180. Menor e posição. Use ponteiro pelo menos uma vez no código.

### Observações:

- É obrigatório usar recursão e ponteiro nos casos indicados
- Será utilizado um programa especializado para detecção de plágio em todas as submissões.

```
#include <stdio.h>
void pares(int n){
    if (n == 0){
        return;
    } else {
        if ((102 - n) \% 2 == 0){
            printf("%d\n", 102 - n);
            pares(n - 2);
int main() {
    pares(100);
    return 0;
```

#### #include <stdio.h>

```
void pares(int n) {
    if (n<=100) {
        printf("%d\n", n);
        pares(n+2);
    }
}</pre>
```

```
int main() {
    pares(2);
    return 0;
}
```

```
#include <stdio.h>
void eh_div(int i, int j){
    if (i == j){
        printf("%d\n", j);
        return;
    } else {
        if (j \% i == 0){
            printf("%d\n", i);
        return eh_div(i + 1, j);
int main() {
    int n;
    scanf("%d", &n);
    eh_div(1, n);
    return 0;
```

#### #include <stdio.h>

```
void eh_div(int n, int i) {
   if (i<=n) {
     if (n%i==0)
        printf("%d\n", i);
     eh_div(n, i+1);
   }
}</pre>
```

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

   eh_div(n, 1);

   return 0;
}
```

### Exemplo de uso de ponteiros

Escreva um programa que leia um número inteiro positivo **n** seguido de **n** números inteiros e imprima esses **n** números em ordem invertida.

Por exemplo, ao receber

5

22 33 44 55 66

o seu programa deve imprimir

66 55 44 33 22

- Seu programa não deve impor limitações sobre o valor de n
- Seu programa não deve usar colchetes.

### Exemplo de uso de ponteiros

```
#include <stdio.h>
   #include <stdlib.h>
 3
   int main() {
 5
        int n, i;
        scanf("%d", &n);
 6
        int *p = (int *) malloc(n*sizeof(int));
        for (i=0; i<n; i++)
10
            scanf("%d", p+i);
11
12
        for (i=n-1; i>=0; i--)
            printf("%d ", *(p+i));
13
14
15
        free(p);
16 }
```

```
5
11 22 33 44 55
55 44 33 22 11
```

Os ponteiros facilitam a alocação dinâmica de memória