

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

Alternativa ONLINE para o Dev C++

- <https://www.onlinegdb.com/>

The screenshot displays the OnlineGDB website interface. The top navigation bar includes the OnlineGDB logo, a 'GDB online Debugger | Compile' tab, and a browser address bar showing 'https://www.onlinegdb.com'. The left sidebar contains navigation links: 'IDE', 'My Projects', 'Learn Programming', 'Programming Questions', 'Sign Up', and 'Login'. Below these are social media icons for Facebook, Twitter, and Google+, along with a '+ 24.3K' badge. A promotional banner for 'monday.com' is also visible. The main content area features a code editor with a C++ program. The code includes headers for `<stdio.h>` and `<stdlib.h>`, and implements a `main()` function that uses `malloc` to allocate memory for arrays `T` and `N`, and a `scanf` to read an integer `T`. A nested loop structure is used to calculate the sum of elements in `N` up to index `T`, with the result printed using `printf`. The bottom of the interface shows execution controls: 'Run', 'Debug', 'Stop', 'Share', 'Save', and 'Beautify' buttons. Below these are fields for 'Command line arguments:' and 'Standard Input:' with radio buttons for 'Interactive Console' (selected) and 'Text'. The footer contains links for 'About', 'FAQ', 'Blog', 'Terms of Use', 'Contact Us', 'GDB Tutorial', and 'Credits', along with the copyright notice '2018 © GDB Online'.

```
7
8 *****/
9 #include<stdio.h>
10 #include<stdlib.h>
11
12 int main()
13 {
14
15     int *T, *N, i, j;
16
17     N = (int*)malloc(1000*sizeof(int));
18     T = (int*)malloc(sizeof(int));
19
20     scanf("%d", T);
21
22     for(i=0; i<1000; i+=*T)
23     {
24         j=0;
25         while(i+j<1000 && j<*T)
26         {
27             *(N+i+j) = j;
28             printf("N[%d] = %d\n", i+j, *(N+i+j));
29             j++;
30         }
31     }
32
33 }
```

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.

Problema 1059

```
1  #include <stdio.h>
2
3  void pares(int n){
4      if (n == 0){
5          return;
6      } else {
7          if ((102 - n) % 2 == 0){
8              printf("%d\n", 102 - n);
9              pares(n - 2);
10         }
11     }
12 }
13
14 int main() {
15     pares(100);
16     return 0;
17 }
```

1059

Problema 1059

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

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

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

1059

Problema 1157

```
1  #include <stdio.h>
2
3  void eh_div(int i, int j){
4      if (i == j){
5          printf("%d\n", j);
6          return;
7      } else {
8          if (j % i == 0){
9              printf("%d\n", i);
10             }
11             return eh_div(i + 1, j);
12         }
13     }
14
15     int main() {
16
17         int n;
18         scanf("%d", &n);
19         eh_div(1, n);
20
21         return 0;
22     }
```

1157

Problema 1157

```
#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);
    }
}
```

```
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

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      int n, i;
6      scanf("%d", &n);
7      int *p = (int *) malloc(n*sizeof(int));
8
9      for (i=0; i<n; i++)
10         scanf("%d", p+i);
11
12     for (i=n-1; i>=0; i--)
13         printf("%d ", *(p+i));
14
15     free(p);
16 }
```

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

Os ponteiros
facilitam a
alocação dinâmica
de memória