INSTITUTO FEDERAL
RIO GRANDE DO SUL

Estrutura de Dados II

Procedimentos e Funções: RECURSIVIDADE

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

INSTITUTO FEDERAL
RIO GRANDE DO SUL

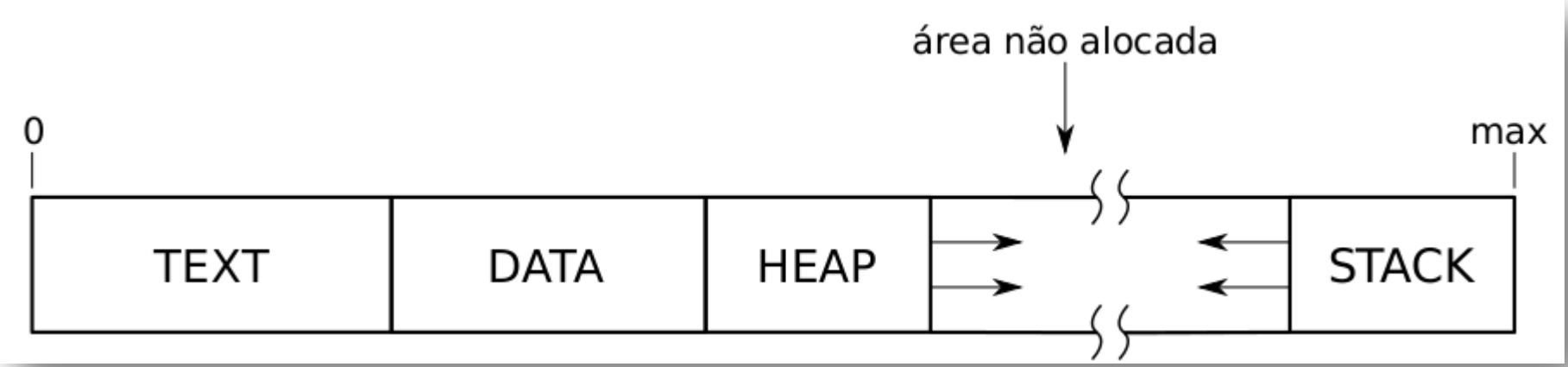
Recursividade

- Um programa recursivo é mais rápido que um programa Interativo?
- É mais fácil programar um programa recursivo que um programa do que interativo?
- Então porque usamos recursividade?

INSTITUTO FEDERAL
RIO GRANDE DO SUL

Recursividade

O espaço de endereços de um processo em execução é dividido em várias segmentos lógicos.



INSTITUTO FEDERAL
RIO GRANDE DO SUL

Recursividade

Text:	• contém o código do programa e suas constantes. Este segmento é alocado durante a criação do processo (''exec'') e permanece do mesmo tamanho durante toda a vida do processo.
Data:	• este segmento é a memória de trabalho do processo, aonde ficam alocadas as variáveis globais e estáticas. Tem tamanho fixo ao longo da execução do processo.
Stack:	• contém a pilha de execução, onde são armazenadas os parâmetros, endereços de retorno e variáveis locais de funções. Pode variar de tamanho durante a execução do processo.
Heap:	• contém blocos de memória alocadas dinamicamente, a pedido do processo, durante sua execução. Varia de tamanho durante a vida do processo.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

Recursividade

A **alocação estática** ocorre quando são declaradas variáveis globais ou estáticas; Alocadas em *Data*.

A **alocação automática** ocorre quando são declaradas variáveis locais e parâmetros de funções. O espaço para a alocação dessas variáveis é reservado quando a função é invocada, e liberado quando a função termina. Usada a pilha (*stack*).

A **alocação dinâmica**, quando o processo requisita explicitamente um bloco de memória para armazenar dados. Usa a área de *heap*.

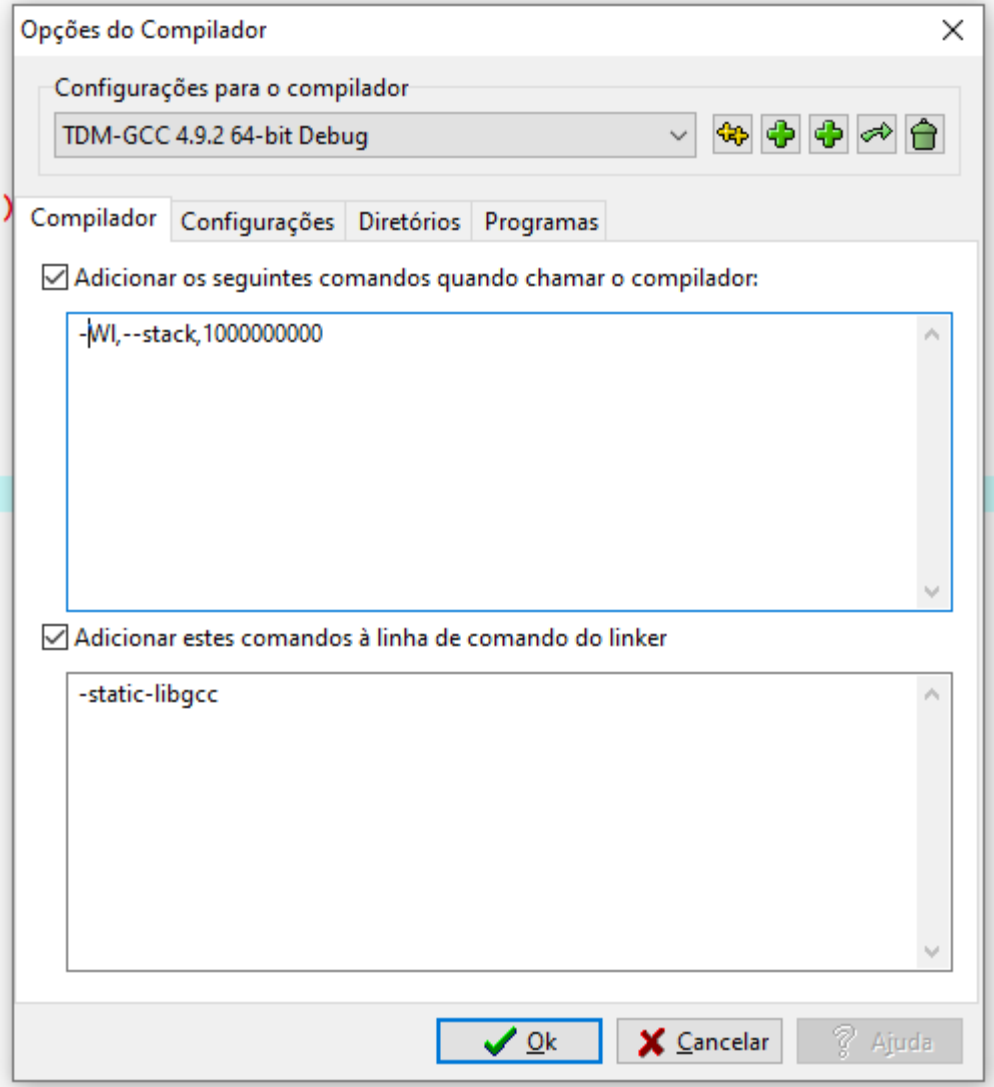
INSTITUTO FEDERAL
RIO GRANDE DO SUL

Recursividade

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

bool imprime_pares(int x, int y)
{


if (x > y)
        return (false);
    if (x == y)
        return (true);
    else if (x % 2 == 0)
        printf ("Valor %d eh par \n", x);

    return imprime_pares (x + 1, y);
    }

main()
{
    if (imprime_pares(100 ,150000))
        printf("Valores impressos");
    else
        printf("erro de função");
}
```

```
C:\Users\lange\Dropbox\Aulas\Aulas\2024\SSI - EstrututaDeDados II\Aula 11 - Recursividade\
Valor 43238 eh par
Valor 43240 eh par
Valor 43242 eh par
Valor 43244 eh par
Valor 43246 eh par
Valor 43248 eh par
Valor 43250 eh par
Valor 43252 eh par
Valor 43254 eh par
Valor 43256 eh par
Valor 43258 eh par
Valor 43260 eh par
Valor 43262 eh par
Valor 43264 eh par
Valor 43266 eh par
Valor 43268 eh par
Valor 43270 eh par
Valor 43272 eh par
Valor 43274 eh par
Valor 43276 eh par

--------------------------------
Process exited after 6.827 seconds with return value 3221225725
Pressione qualquer tecla para continuar. . .
```

Recursividade



-Wl,--stack,1000000000

INSTITUTO FEDERAL
RIO GRANDE DO SUL

Recursividade

recursivo.cpp

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  bool imprime_pares(int x, int y)
5  {
6
7
8  if (x > y)
9      return (false);
10     if (x == y)
11         return (true);
12     else if (x % 2 == 0)
13         printf ("Valor %d eh par \n", x);
14
15     return imprime_pares (x + 1, y);
16     }
17
18 main()
19 {
20     if (imprime_pares(100 ,150000))
21         printf("Valores impressos");
22     else
23         printf("erro de função");
24 }
25
```

```
C:\Users\lange\Dropbox\Aulas\Aulas\2024\SSI - EstrututaDeDados II\Aula 11 - Recu
Valor 149948 eh par
Valor 149950 eh par
Valor 149952 eh par
Valor 149954 eh par
Valor 149956 eh par
Valor 149958 eh par
Valor 149960 eh par
Valor 149962 eh par
Valor 149964 eh par
Valor 149966 eh par
Valor 149968 eh par
Valor 149970 eh par
Valor 149972 eh par
Valor 149974 eh par
Valor 149976 eh par
Valor 149978 eh par
Valor 149980 eh par
Valor 149982 eh par
Valor 149984 eh par
Valor 149986 eh par
Valor 149988 eh par
Valor 149990 eh par
Valor 149992 eh par
Valor 149994 eh par
Valor 149996 eh par
Valor 149998 eh par
Valores impressos
--------------------------------
Process exited after 10.3 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

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

bool imprime_pares(int x, int y)
{
if (x > y)
                return (false);
        if (x == y)
                return (true);
        else if (x % 2 == 0)
                printf ("Valor %d eh par \n", x);

        return imprime_pares(x + 1, y);
        }
main()
{
        if (imprime_pares(100 ,150000))
                printf("Valores impressos");
        else
                printf("erro de função");
}
```