Construção de Compiladores
Período Especial
Aula 8: Comando While

Bruno Müller Junior

Departamento de Informática
UFPR

2020

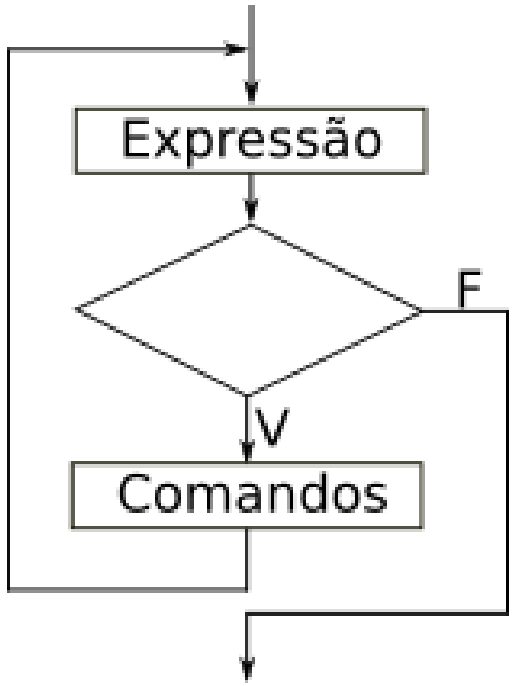
Sintaxe

- O comando `while` (regra 23) é uma das possibilidades de `<comando sem rótulo>` (regra 18).
- O analisador sintático irá escolher a regra 18 quando estiver na regra 23 e encontrar o token `while`.

```
18.  <comando sem rótulo> ::= <atribuição>          |
                              ...                   |
                              <comando repetitivo>
23. <comando repetitivo>  ::= WHILE <expressão> DO
                              <comando sem rótulo>
```

Fluxo

- O fluxo de execução do comando while não é sequencial.
- Existem desvios do fluxo para locais específicos.
- Os desvios são de dois tipos: incondicionais e condicionais.



Desvio na MEPA

- Para implementar a mudança de fluxo, a MEPA contém duas instruções:

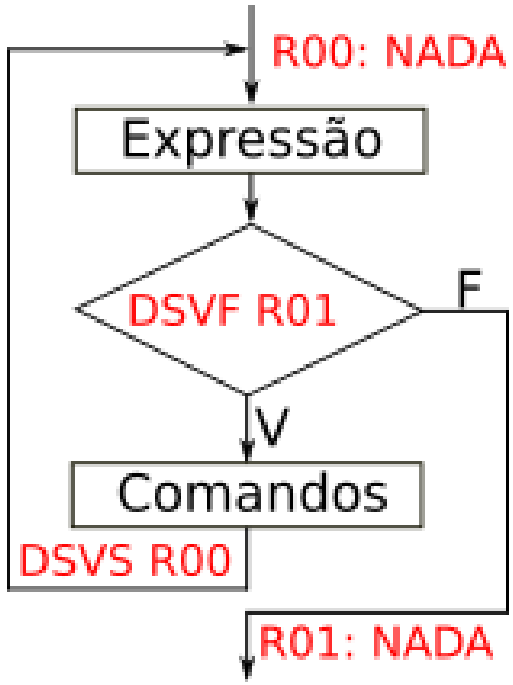
Instrução	Ação	Significado
DSVS p	`i:=p;`	Desvia Sempre

DSVF p	`Se (M[s]==0) então i:=p;`
	`senão i:=i+1`
	`s:=s-1`
		Desvia se Falso

- Em ambas o parâmetro é o endereço de desvio do fluxo, chamado `rótulo`.
- Por convenção deste material, os rótulos seguem sempre o padrão da letra `Rdd`, onde dd são dois dígitos. Exemplos: R00, R01, ...

Modelo

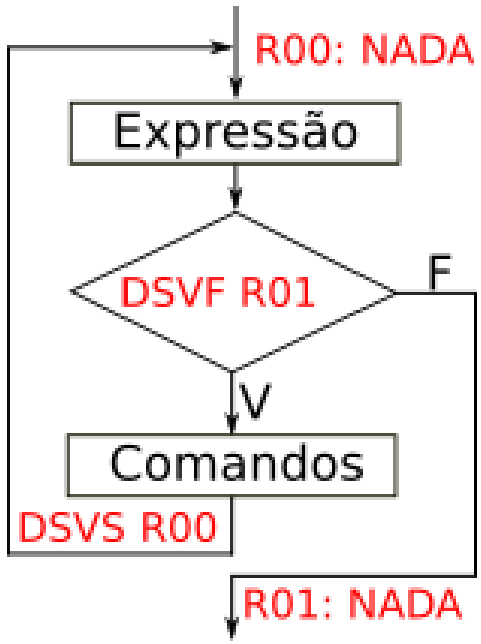
- A tradução do programa para MEPA é simples:
  - inclui-se rótulos nas entradas de fluxo;
  - inclui-se desvios condicionais analisando o resultado da expressão.
  - inclui-se desvios incondicionais ao final do comando.



Esquema de Tradução

- Outra forma de ver é através do esquema de tradução;

WHILE		R00: NADA
E		*Traduz Expressão* DSVF R01
DO		
<comandos>		*Traduz Comandos* DSVS R00 R01: NADA

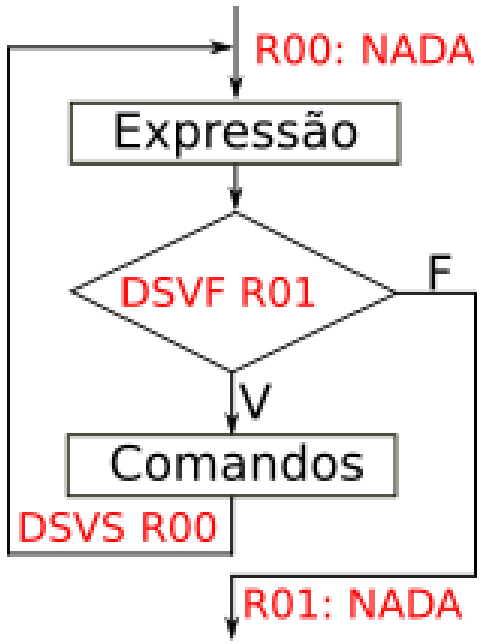


Esquema de Tradução

- Que já sugere a geração de código:

```
23. <comando repetitivo>  ::= WHILE <expressão> DO <comando sem rótulo>
```

WHILE	R00: NADA
E	*Traduz Expressão* DSVF R01
DO	
<comandos>	*Traduz Comandos* DSVS R00 R01: NADA



Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

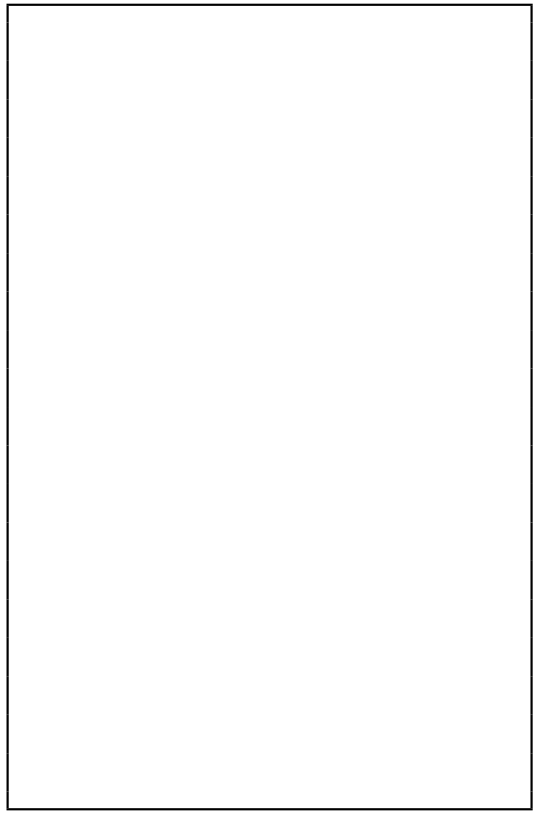
Exemplo

```
program comandoWhile;                    INPP
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

INPP



Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
INPP
```

```
  n      VS    [0,0,?]
Símb.   Cat.  Infos
```

Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
INPP
```

```
  k       VS    [0,1,?]
  n       VS    [0,0,?]
Símb.    Cat.   Infos
```

Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
INPP
AMEM 2
```

```
  k      VS    [0,1,int]
  n      VS    [0,0,int]
Símb.   Cat.   Infos
```

Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
INPP
AMEM 2
AMEM 3
```

Símb.	Cat.	Infos
n	VS	[0,0,int]
k	VS	[0,1,int]
f1	VS	[0,2,int]
f2	VS	[0,3,int]
f3	VS	[0,4,int]

Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
INPP
AMEM 2
AMEM 3
```

```
  f3      VS    [0,4,int]
  f2      VS    [0,3,int]
  f1      VS    [0,2,int]
  k       VS    [0,1,int]
  n       VS    [0,0,int]
Símb.   Cat.   Infos
```

Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
INPP
AMEM 2
AMEM 3
LEIT
ARMZ 0,0
```

```
  f3      VS    [0,4,int]
  f2      VS    [0,3,int]
  f1      VS    [0,2,int]
  k       VS    [0,1,int]
  n       VS    [0,0,int]
Símb.   Cat.   Infos
```

Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
INPP
AMEM 2
AMEM 3
LEIT
ARMZ 0,0
CRCT 0
ARMZ 0,2
```

```
  f3      VS    [0,4,int]
  f2      VS    [0,3,int]
  f1      VS    [0,2,int]
  k       VS    [0,1,int]
  n       VS    [0,0,int]
Símb.   Cat.   Infos
```

Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
INPP
AMEM 2
AMEM 3
LEIT
ARMZ 0,0
CRCT 0
ARMZ 0,2
CRCT 1
ARMZ 0,3
```

```
  f3      VS    [0,4,int]
  f2      VS    [0,3,int]
  f1      VS    [0,2,int]
  k       VS    [0,1,int]
  n       VS    [0,0,int]
Símb.   Cat.   Infos
```

Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
INPP
AMEM 2
AMEM 3
LEIT
ARMZ 0,0
CRCT 0
ARMZ 0,2
CRCT 1
ARMZ 0,3
CRCT 1
ARMZ 0,1
```

```
  f3      VS    [0,4,int]
  f2      VS    [0,3,int]
  f1      VS    [0,2,int]
  k       VS    [0,1,int]
  n       VS    [0,0,int]
Símb.   Cat.   Infos
```

Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
      INPP
      AMEM 2
      AMEM 3
      LEIT
      ARMZ 0,0
      CRCT 0
      ARMZ 0,2
      CRCT 1
      ARMZ 0,3
      CRCT 1
      ARMZ 0,1
R00:  NADA
```

Símb.	Cat.	Infos
f3	VS	[0,4,int]
f2	VS	[0,3,int]
f1	VS	[0,2,int]
k	VS	[0,1,int]
n	VS	[0,0,int]

Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
      INPP
      AMEM 2
      AMEM 3
      LEIT
      ARMZ 0,0
      CRCT 0
      ARMZ 0,2
      CRCT 1
      ARMZ 0,3
      CRCT 1
      ARMZ 0,1
R00:  NADA
      CRVL 0,1
      CRVL 0,0
      CMEG
```

```
  f3      VS    [0,4,int]
  f2      VS    [0,3,int]
  f1      VS    [0,2,int]
  k       VS    [0,1,int]
  n       VS    [0,0,int]
Símb.   Cat.   Infos
```

Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
      INPP
      AMEM 2
      AMEM 3
      LEIT
      ARMZ 0,0
      CRCT 0
      ARMZ 0,2
      CRCT 1
      ARMZ 0,3
      CRCT 1
      ARMZ 0,1
R00:  NADA
      CRVL 0,1
      CRVL 0,0
      CMEG
      DSVF R01
```

```
  f3      VS    [0,4,int]
  f2      VS    [0,3,int]
  f1      VS    [0,2,int]
  k       VS    [0,1,int]
  n       VS    [0,0,int]
Símb.   Cat.   Infos
```

Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
      INPP
      AMEM 2
      AMEM 3
      LEIT
      ARMZ 0,0
      CRCT 0
      ARMZ 0,2
      CRCT 1
      ARMZ 0,3
      CRCT 1
      ARMZ 0,1
R00:  NADA
      CRVL 0,1
      CRVL 0,0
      CMEG
      DSVF R01
      CRVL 0,3
      CRVL 0,2
      SOMA
      ARMZ 0,4
```

```
  f3      VS    [0,4,int]
  f2      VS    [0,3,int]
  f1      VS    [0,2,int]
  k       VS    [0,1,int]
  n       VS    [0,0,int]
Símb.   Cat.   Infos
```

Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
      INPP          CRVL 0,3
      AMEM 2        ARMZ 0,2
      AMEM 3
      LEIT          CRVL 0,4
      ARMZ 0,0      ARMZ 0,3
      CRCT 0
      ARMZ 0,2      CRVL 0,1
      CRCT 1        CRCT 1
      ARMZ 0,3      ARMZ 0,1
      CRCT 1
      ARMZ 0,1
R00:  NADA
      CRVL 0,1
      CRVL 0,0
      CMEG
      DSVF R01
      CRVL 0,3
      CRVL 0,2
      SOMA
      ARMZ 0,4
```

```
  f3      VS    [0,4,int]
  f2      VS    [0,3,int]
  f1      VS    [0,2,int]
  k       VS    [0,1,int]
  n       VS    [0,0,int]
Símb.   Cat.   Infos
```

Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
      INPP
      AMEM 2
      AMEM 3
      LEIT
      ARMZ 0,0
      CRCT 0
      ARMZ 0,2
      CRCT 1
      ARMZ 0,3
      CRCT 1
      ARMZ 0,1
R00:  NADA
      CRVL 0,1
      CRVL 0,0
      CMEG
      DSVF R01
      CRVL 0,3
      CRVL 0,2
      SOMA
      ARMZ 0,4
      CRVL 0,3
      ARMZ 0,2
      CRVL 0,4
      ARMZ 0,3
      CRVL 0,1
      CRCT 1
      ARMZ 0,1
      DSVS R00
R01:  NADA
```

```
  f3      VS    [0,4,int]
  f2      VS    [0,3,int]
  f1      VS    [0,2,int]
  k       VS    [0,1,int]
  n       VS    [0,0,int]
Símb.   Cat.   Infos
```

Exemplo

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
      INPP
      AMEM 2
      AMEM 3
      LEIT
      ARMZ 0,0
      CRCT 0
      ARMZ 0,2
      CRCT 1
      ARMZ 0,3
      CRCT 1
      ARMZ 0,1
R00:  NADA
      CRVL 0,1
      CRVL 0,0
      CMEG
      DSVF R01
      CRVL 0,3
      CRVL 0,2
      SOMA
      ARMZ 0,4
      CRVL 0,3
      ARMZ 0,2
      CRVL 0,4
      ARMZ 0,3
      CRVL 0,1
      CRCT 1
      ARMZ 0,1
      DSVS R00
R01:  NADA
      CRVL 0,0
      IMPR
      CRVL 0,1
      IMPR
```

```
  f3      VS    [0,4,int]
  f2      VS    [0,3,int]
  f1      VS    [0,2,int]
  k       VS    [0,1,int]
  n       VS    [0,0,int]
Símb.   Cat.   Infos
```

Exemplo

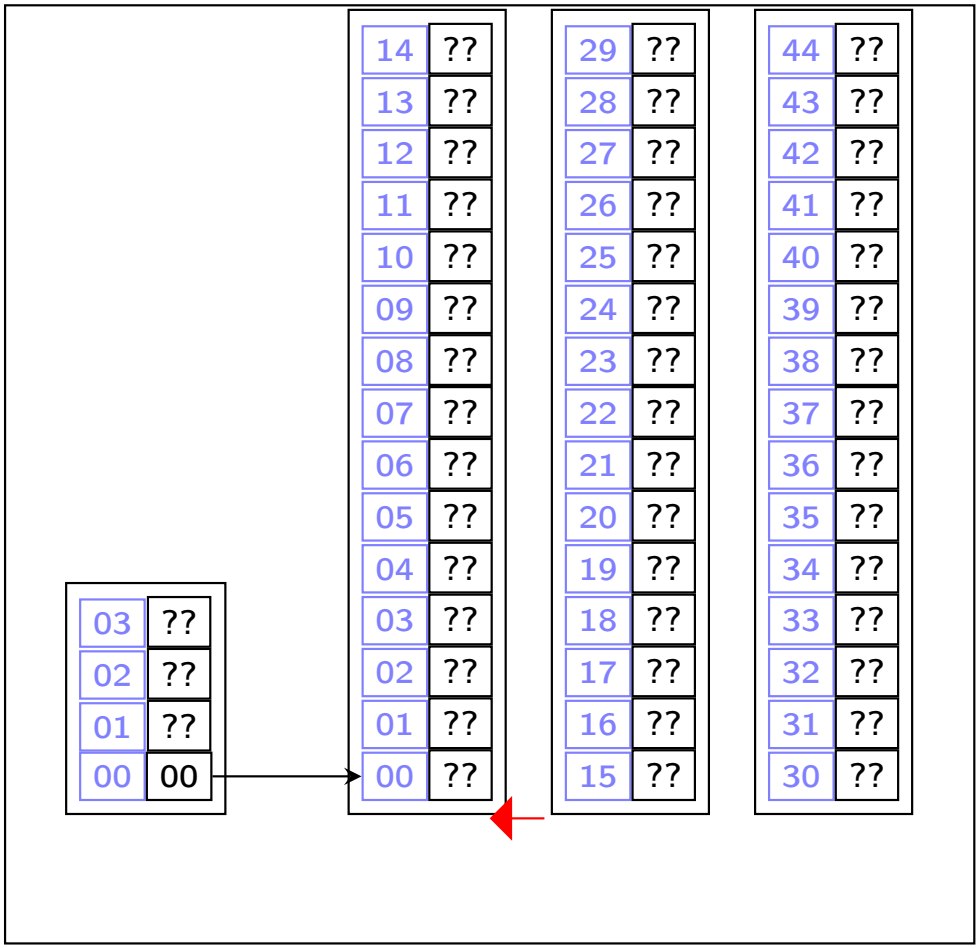
```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```

```
      INPP
      AMEM 2
      AMEM 3
      LEIT
      ARMZ 0,0
      CRCT 0
      ARMZ 0,2
      CRCT 1
      ARMZ 0,3
      CRCT 1
      ARMZ 0,1
R00:  NADA
      CRVL 0,1
      CRVL 0,0
      CMEG
      DSVF R01
      CRVL 0,3
      CRVL 0,2
      SOMA
      ARMZ 0,4
      CRVL 0,3
      ARMZ 0,2
      CRVL 0,4
      ARMZ 0,3
      CRVL 0,1
      CRCT 1
      ARMZ 0,1
      DSVS R00
R01:  NADA
      CRVL 0,0
      IMPR
      CRVL 0,1
      IMPR
      DMEM 5
      PARA
```

Símb.	Cat.	Infos

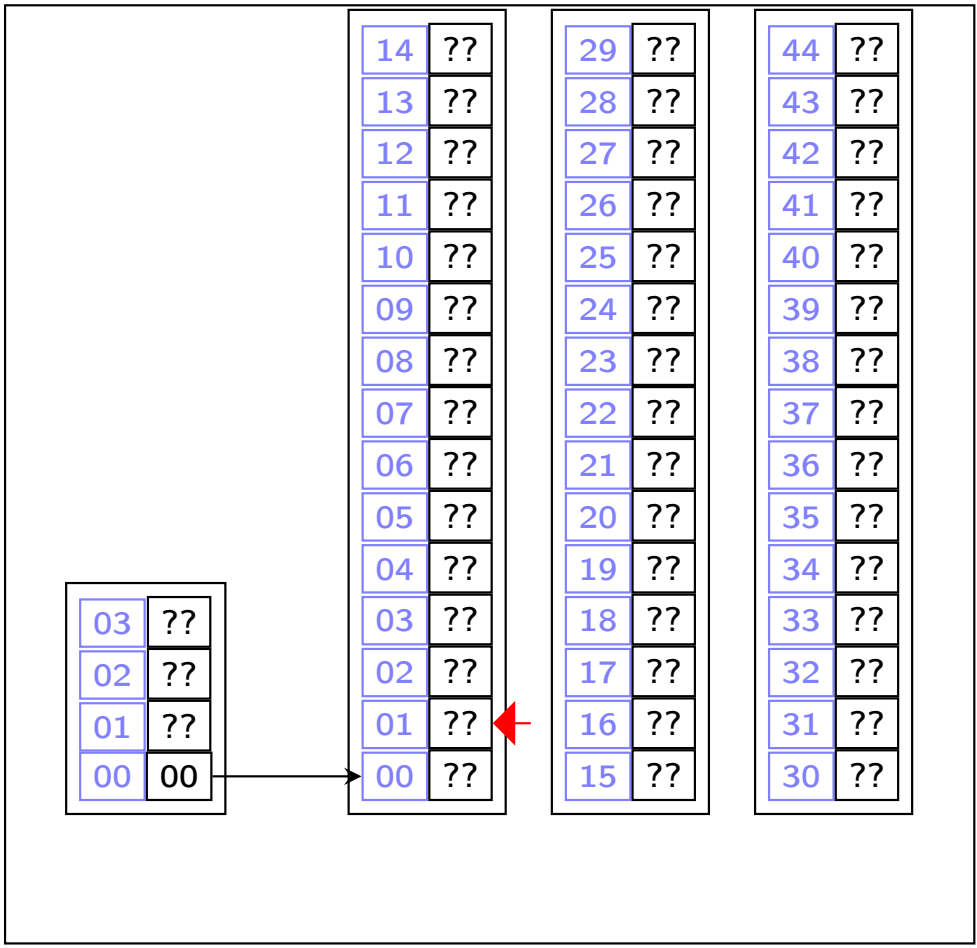
Exec.MEPA

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



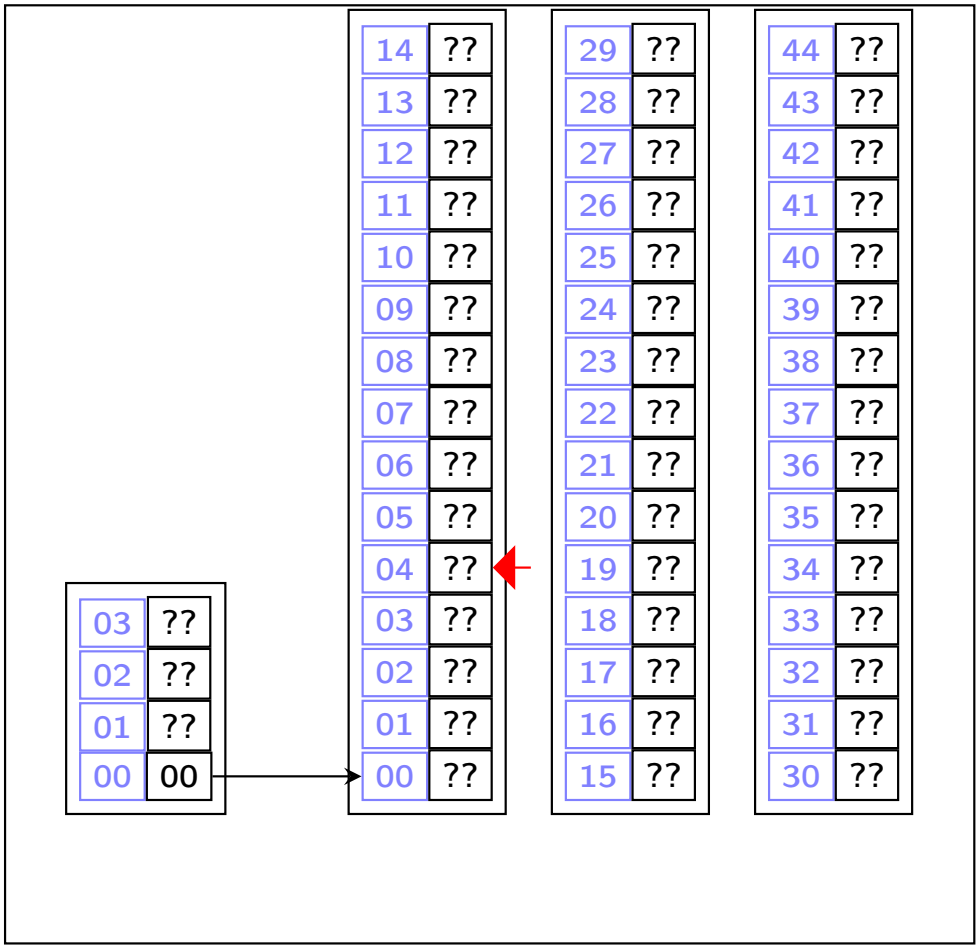
Exec.MEPA

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:  NADA            CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



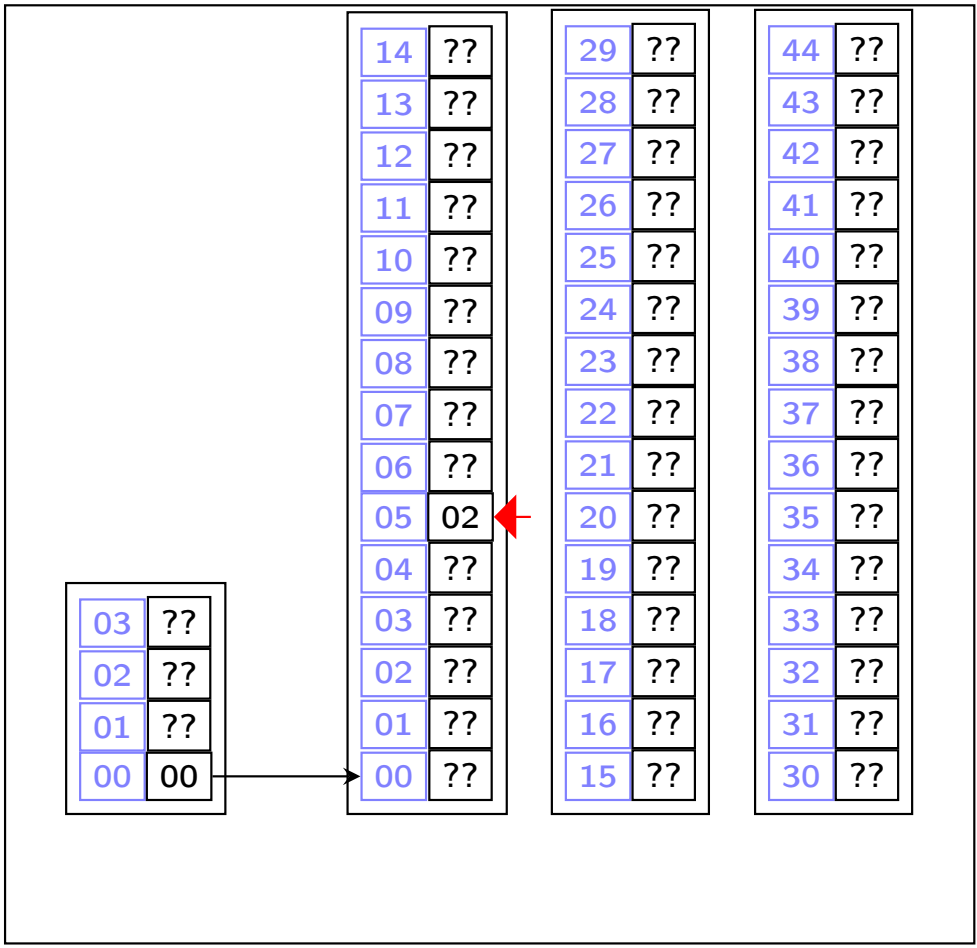
Exec.MEPA

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



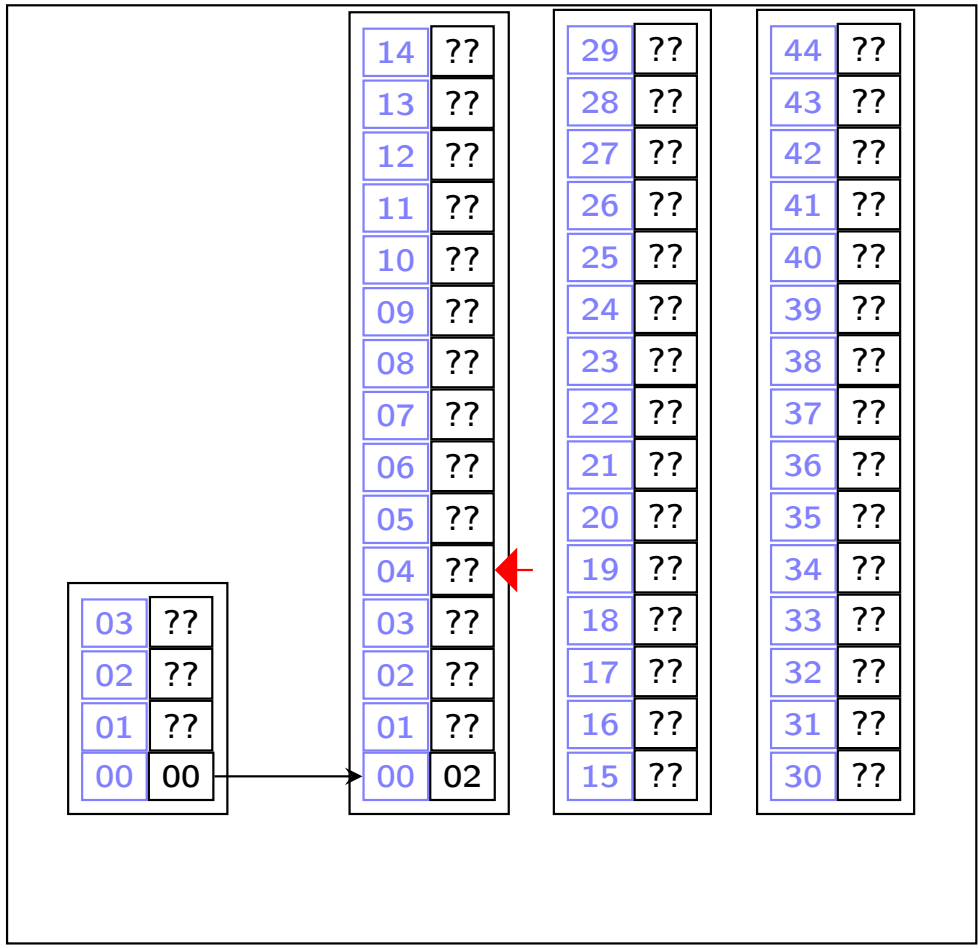
Exec.MEPA

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



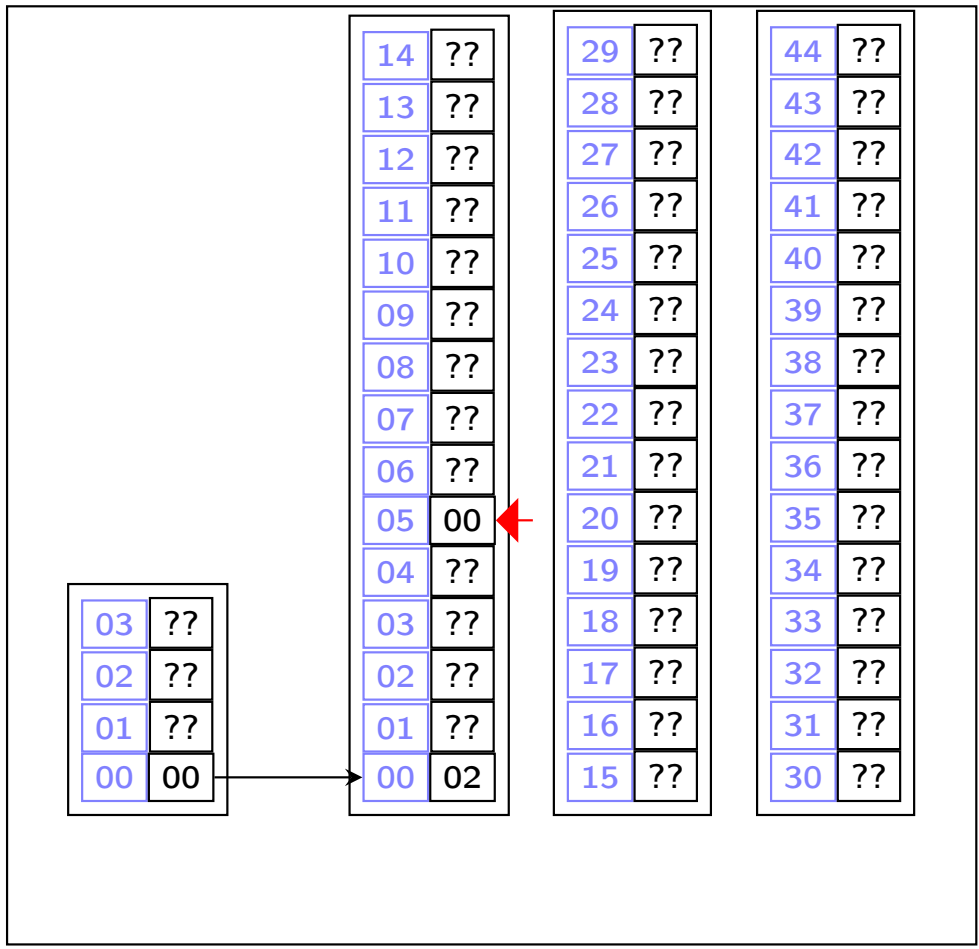
Exec.MEPA

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



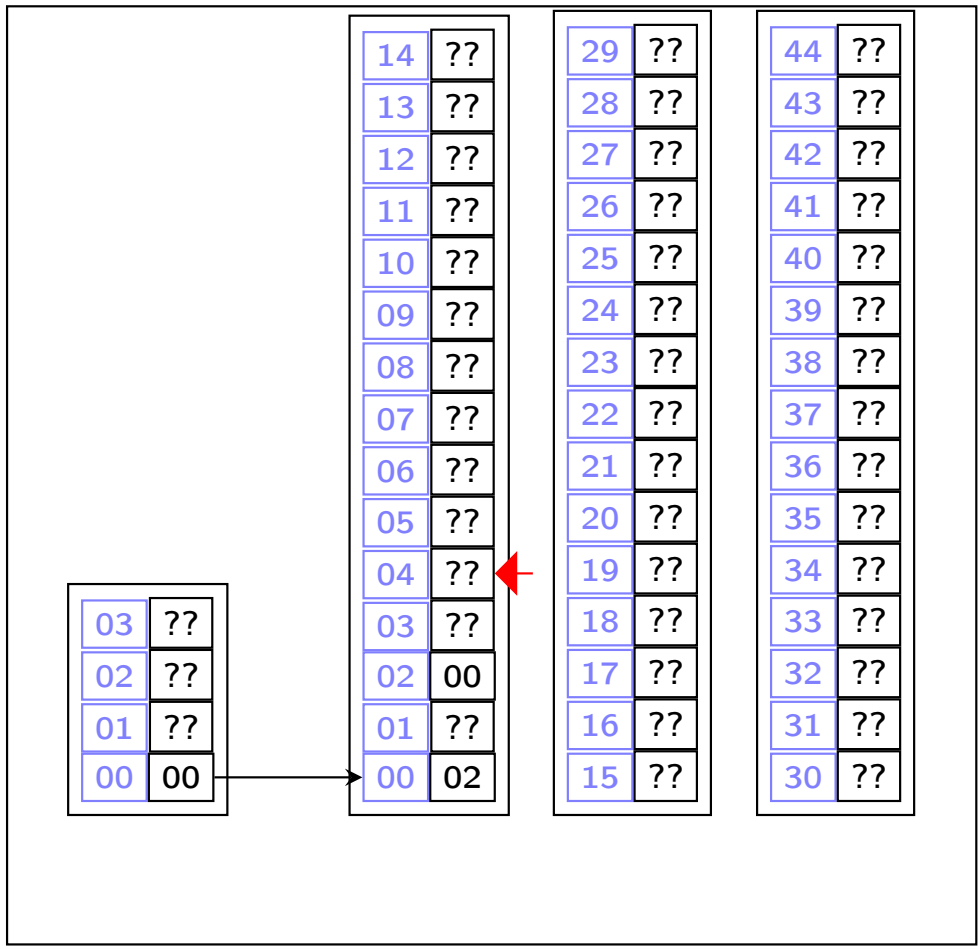
Exec.MEPA

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:  NADA            CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



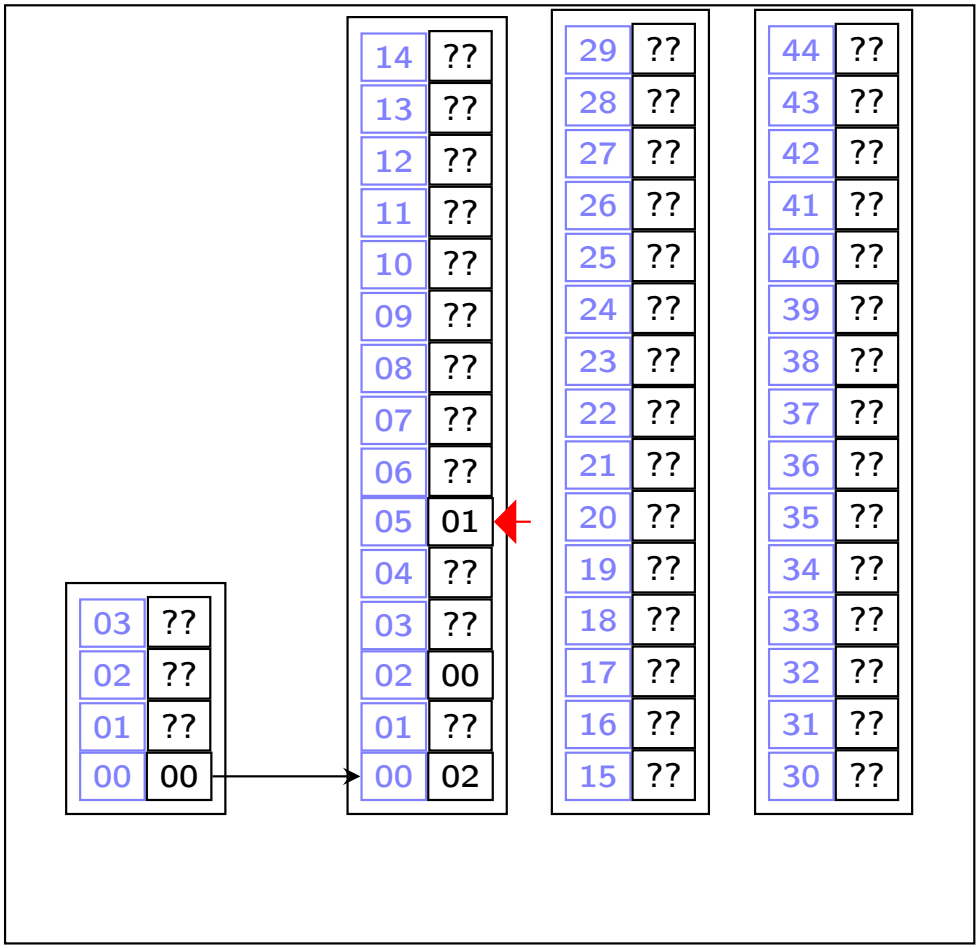
Exec.MEPA

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:  NADA            CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



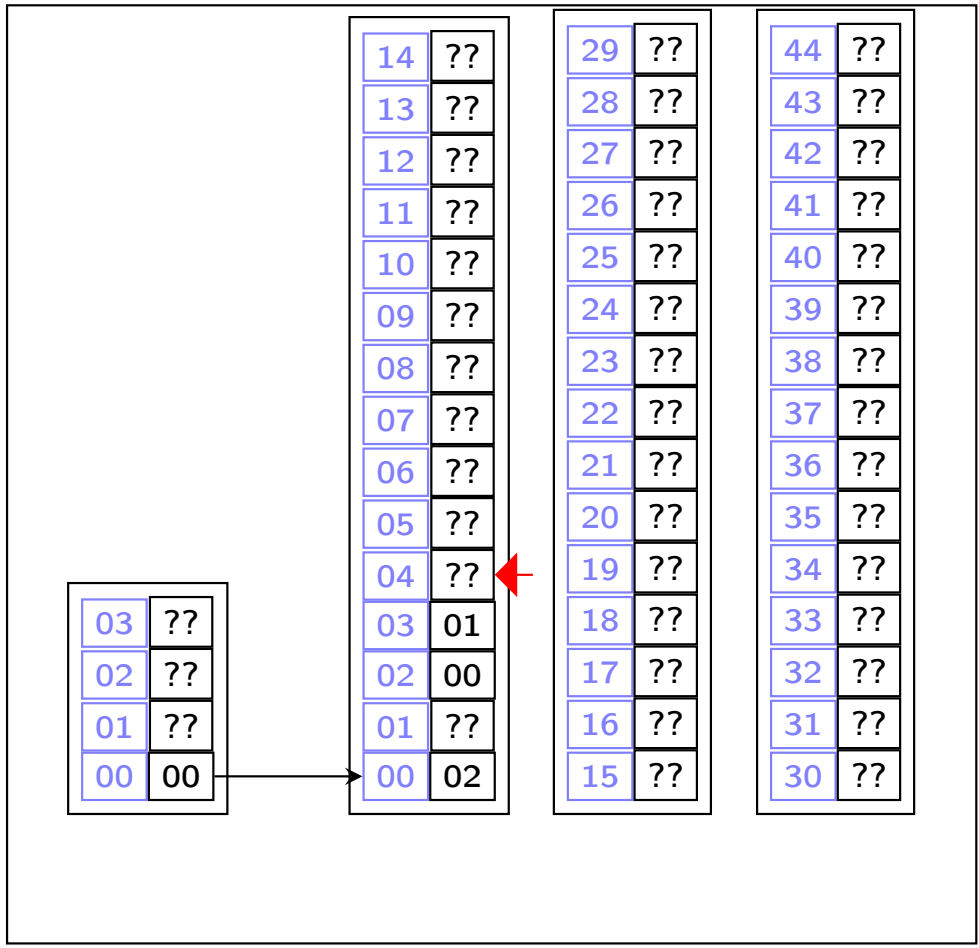
Exec.MEPA

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:  NADA            CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



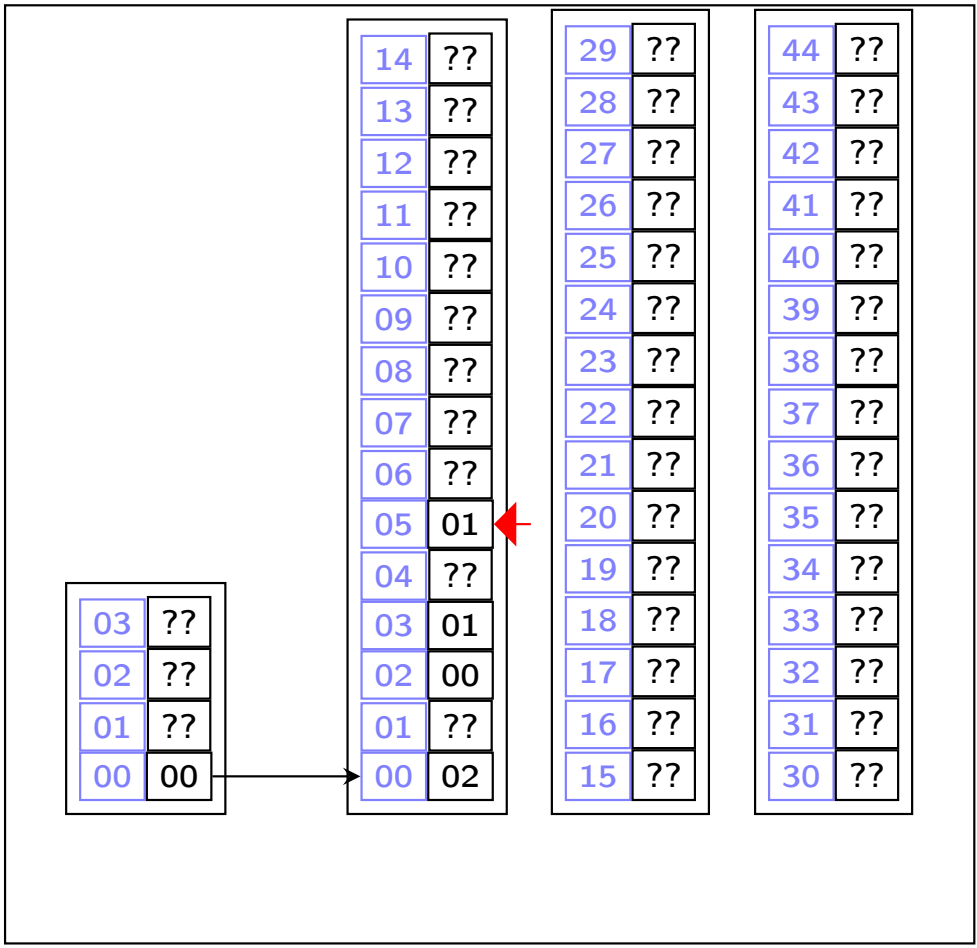
Exec.MEPA

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



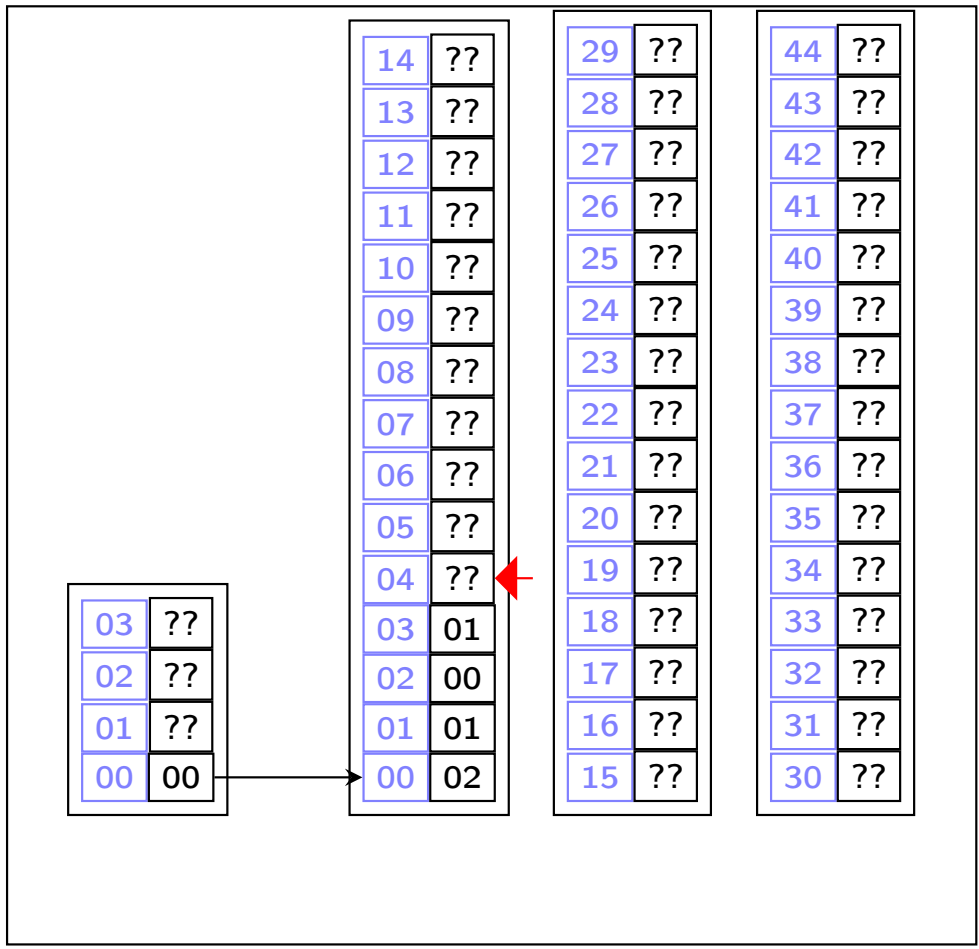
Exec.MEPA

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



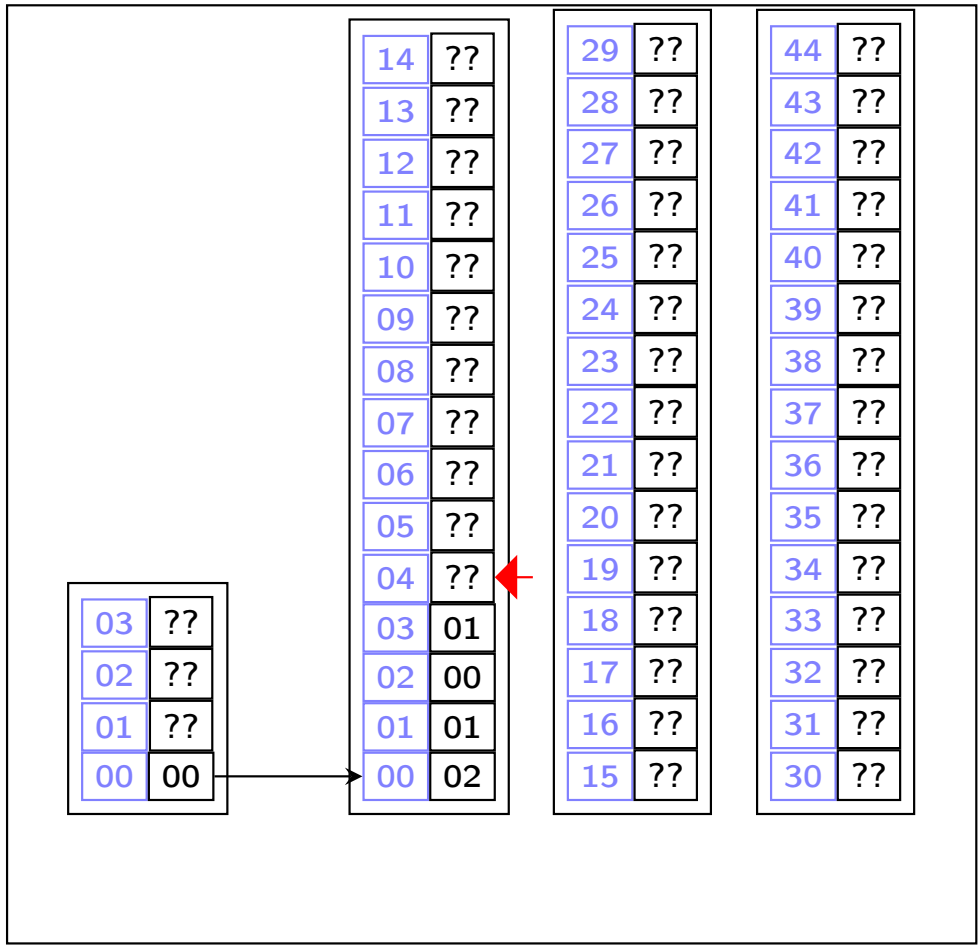
Exec.MEPA

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



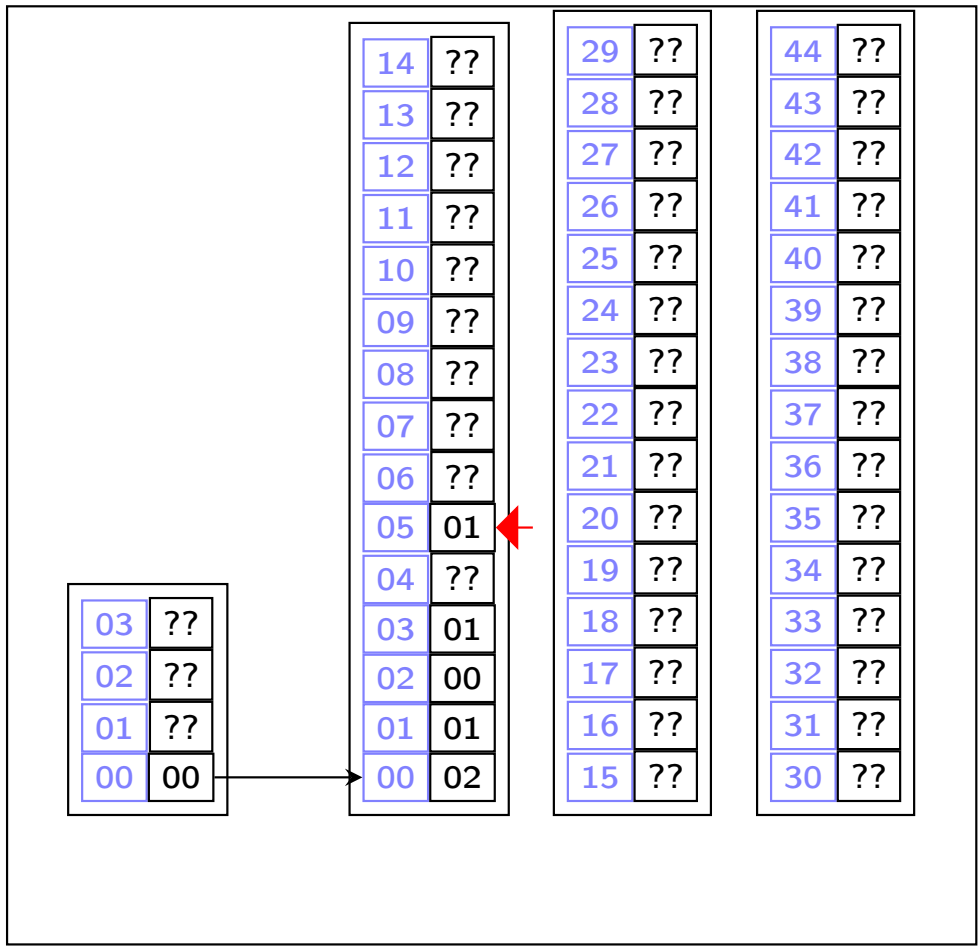
Primeira Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



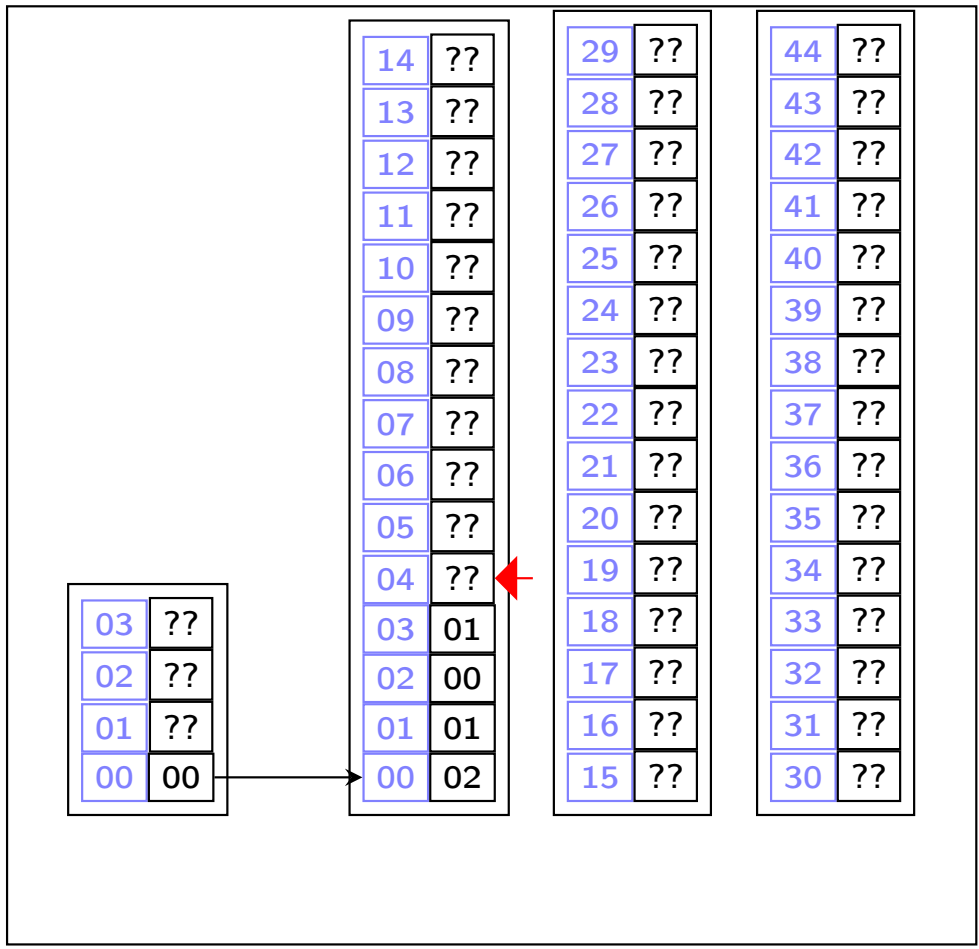
Primeira Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



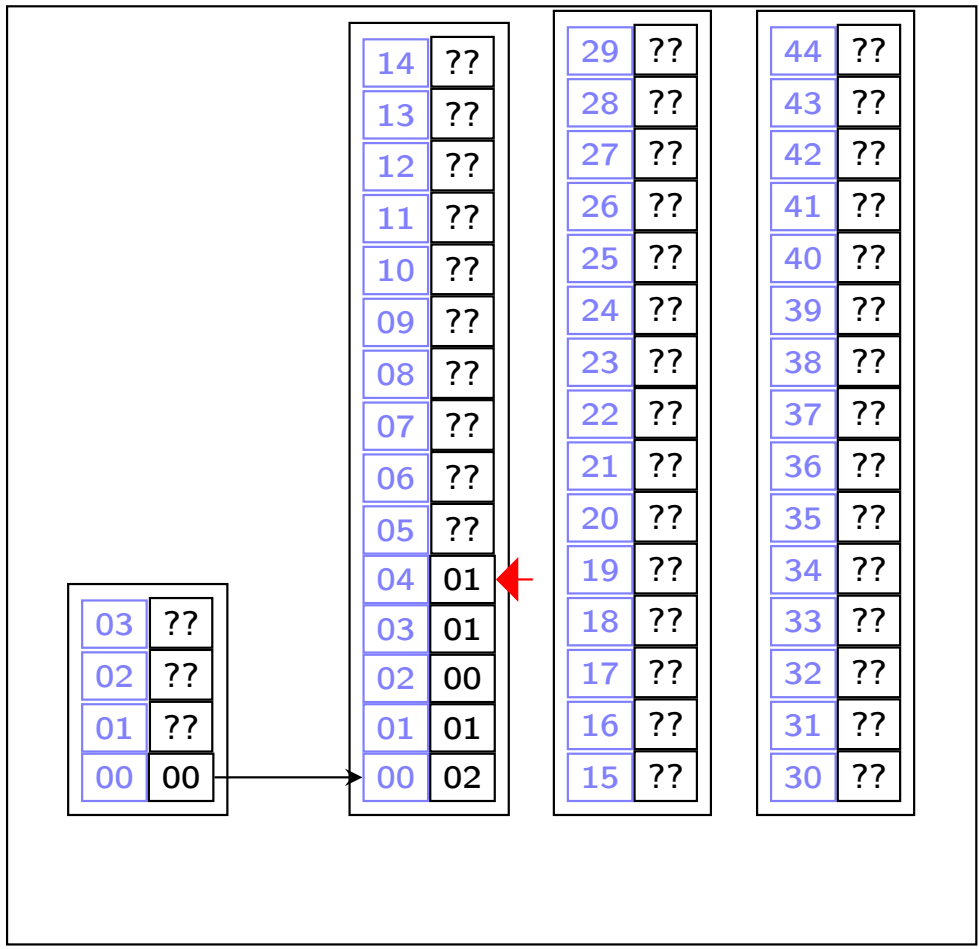
Primeira Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



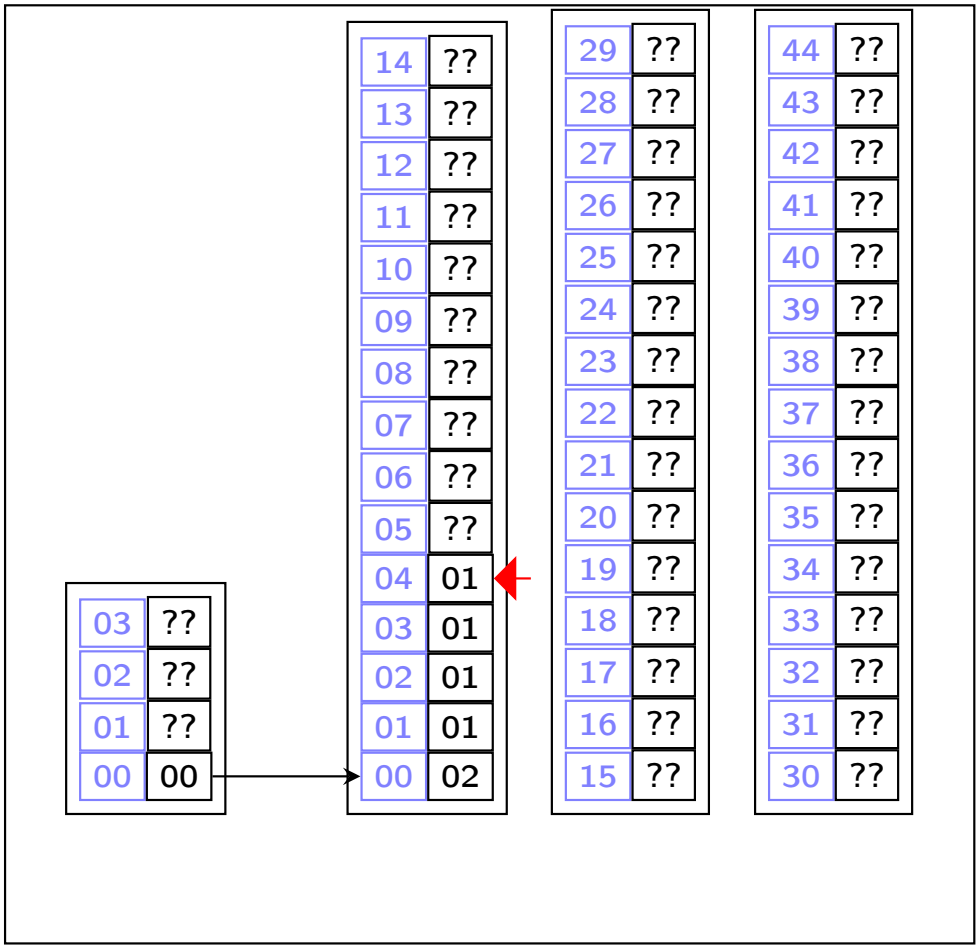
Primeira Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:  NADA            CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



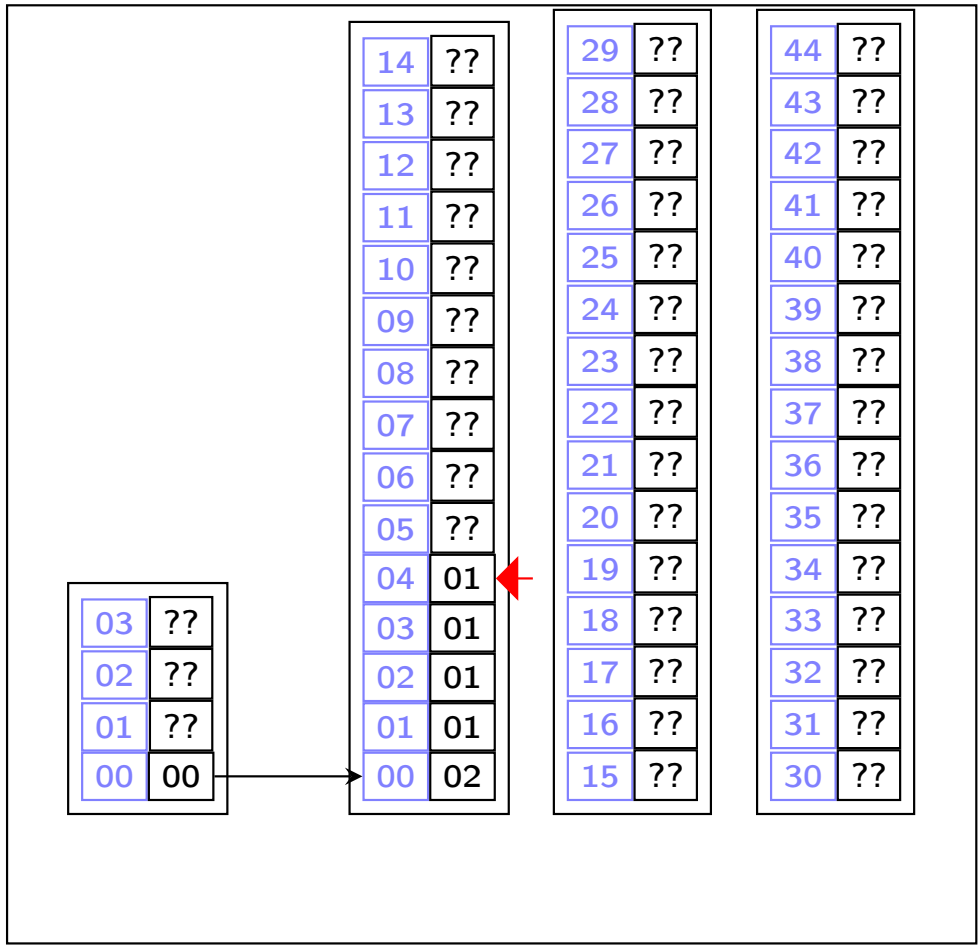
Primeira Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



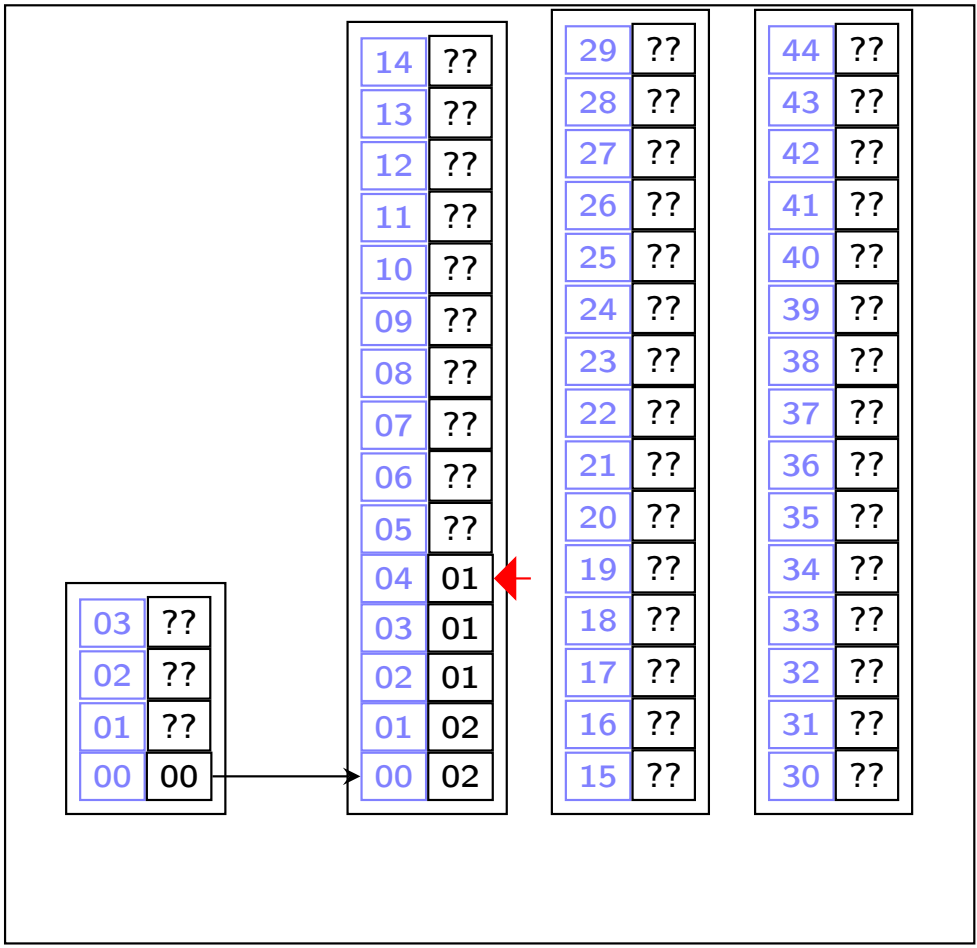
Primeira Iteração

```
     INPP            CRVL 0,3
     AMEM 2          ARMZ 0,2
     AMEM 3
     LEIT            CRVL 0,4
     ARMZ 0,0        ARMZ 0,3
     CRCT 0
     ARMZ 0,2        CRVL 0,1
     CRCT 1          CRCT 1
     ARMZ 0,3        ARMZ 0,1
     CRCT 1          DSVS R00
     ARMZ 0,1   R01: NADA
R00:NADA             CRVL 0,0
     CRVL 0,1        IMPR
     CRVL 0,0        CRVL 0,1
     CMEG            IMPR
     DSVF R01        DMEM 5
     CRVL 0,3        PARA
     CRVL 0,2
     SOMA
     ARMZ 0,4
```



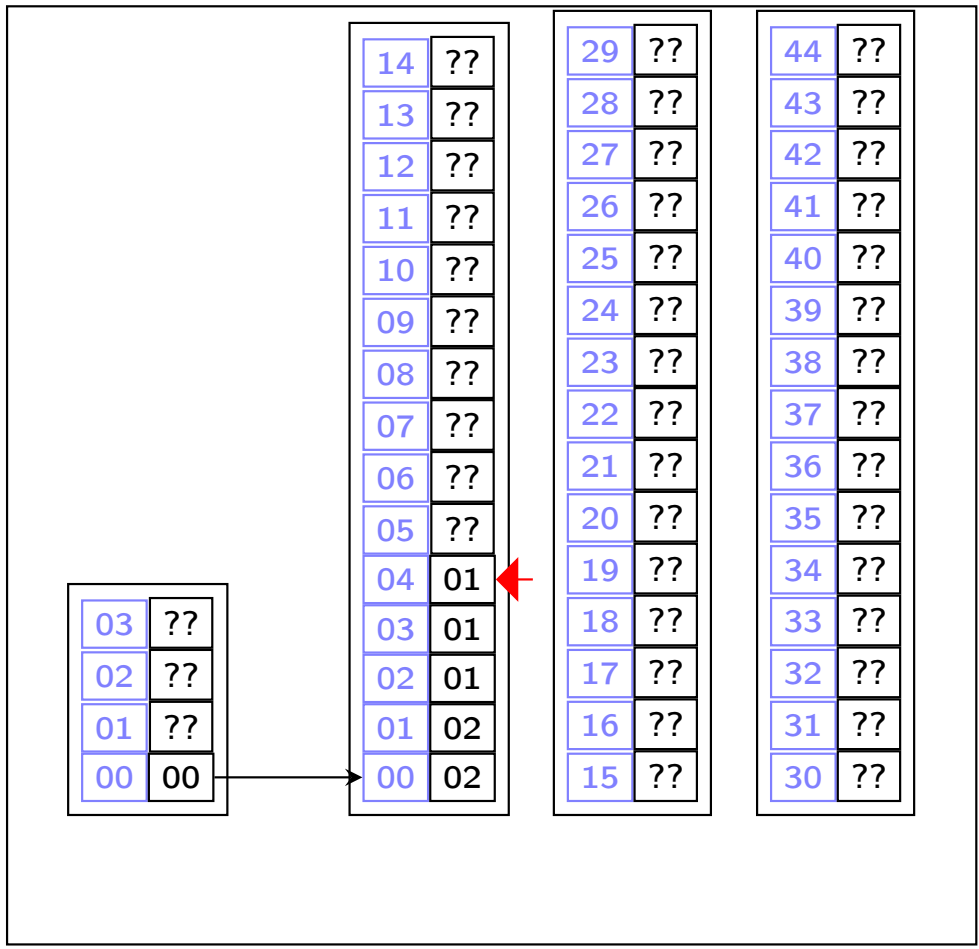
Primeira Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:  NADA            CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



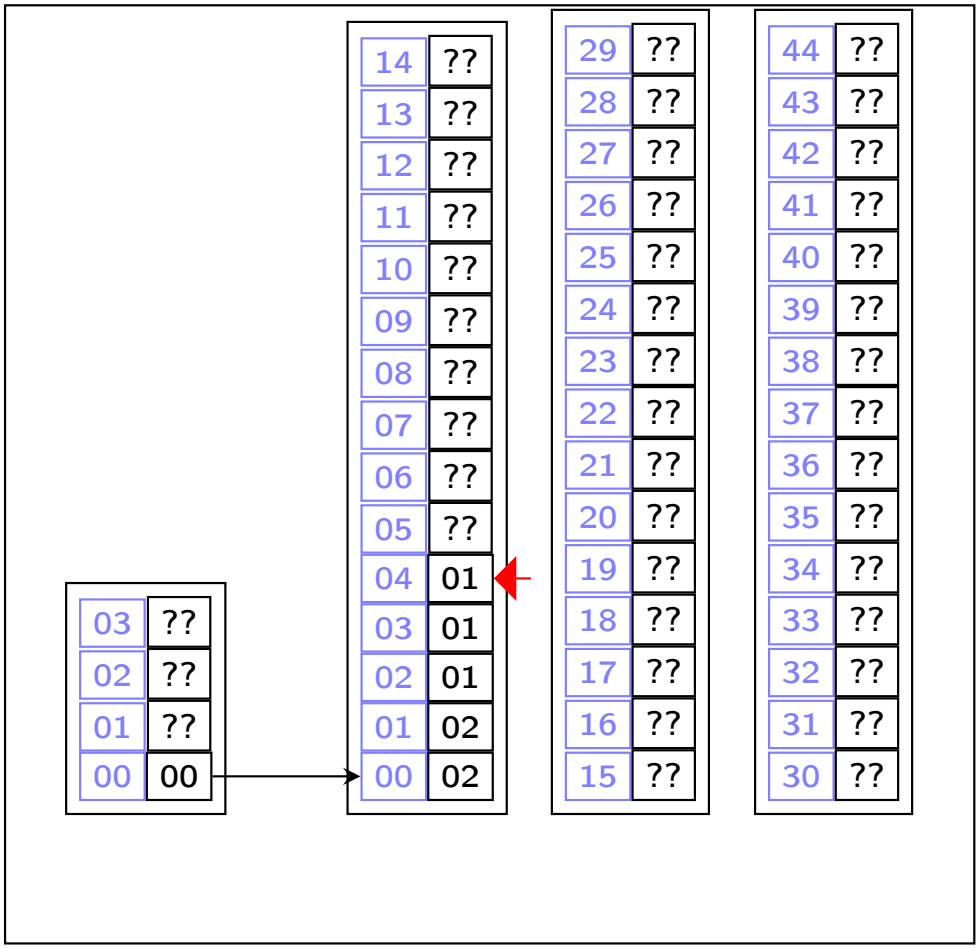
Primeira Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



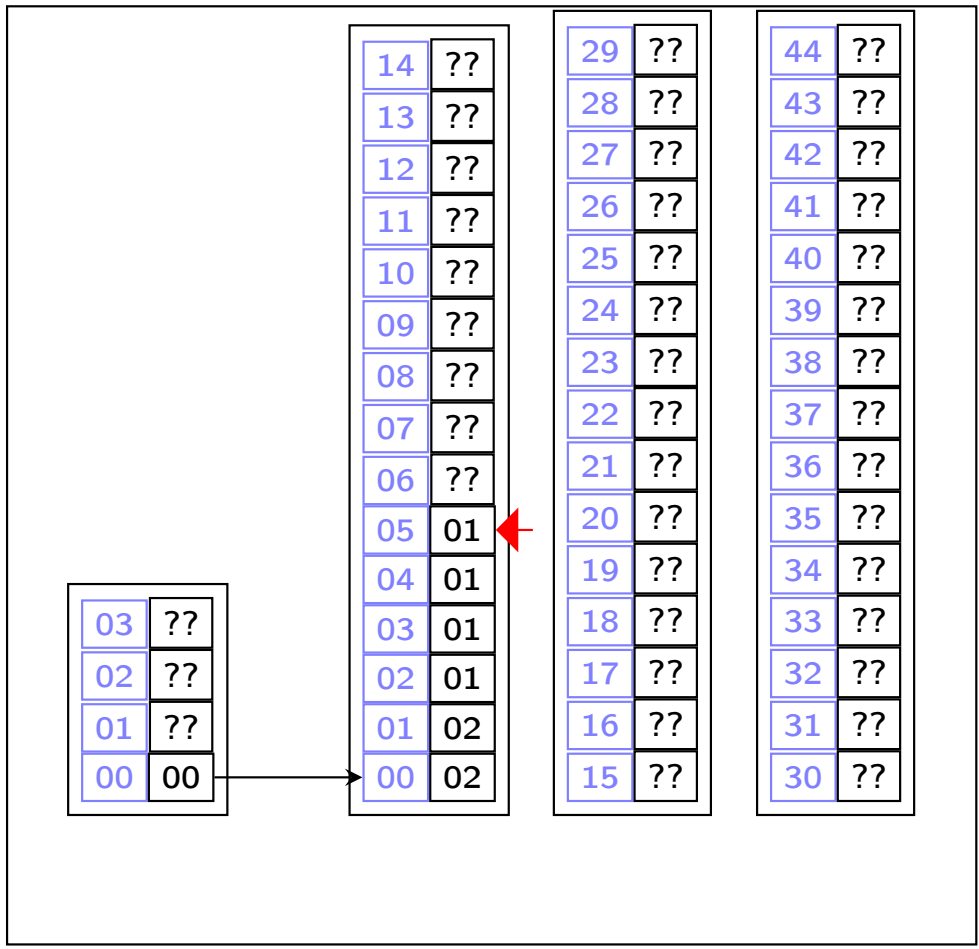
Segunda Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



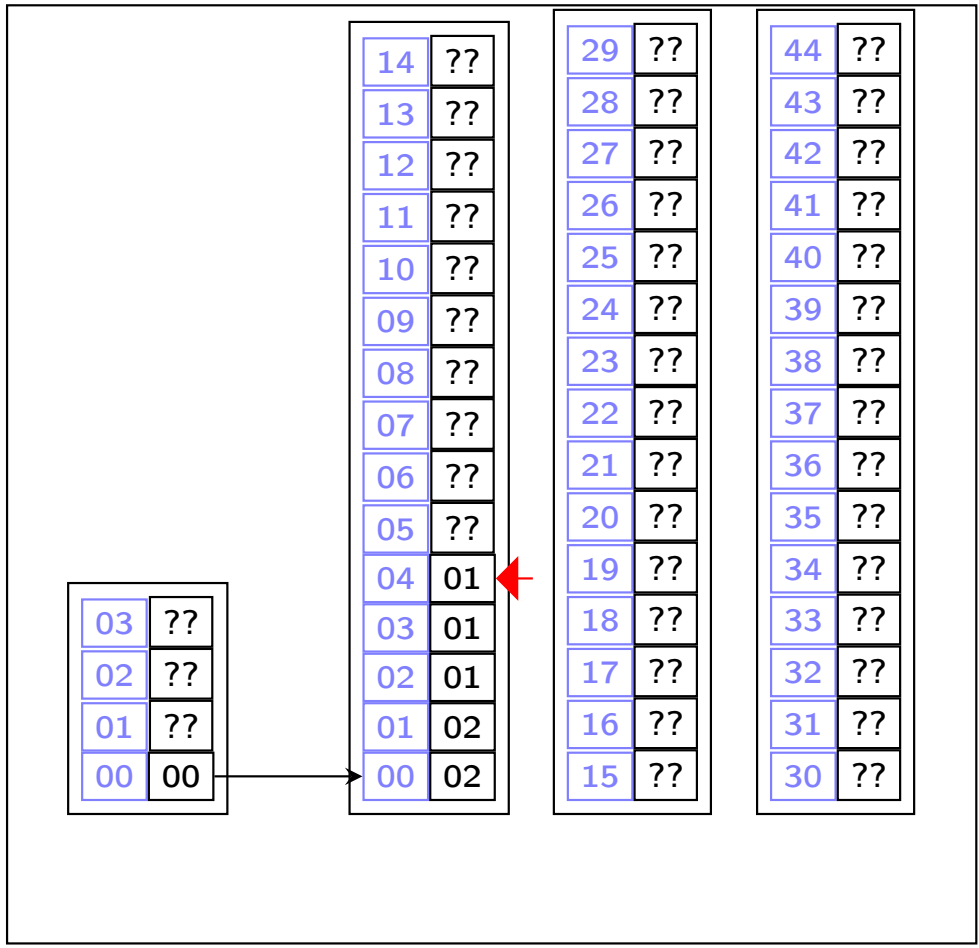
Segunda Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



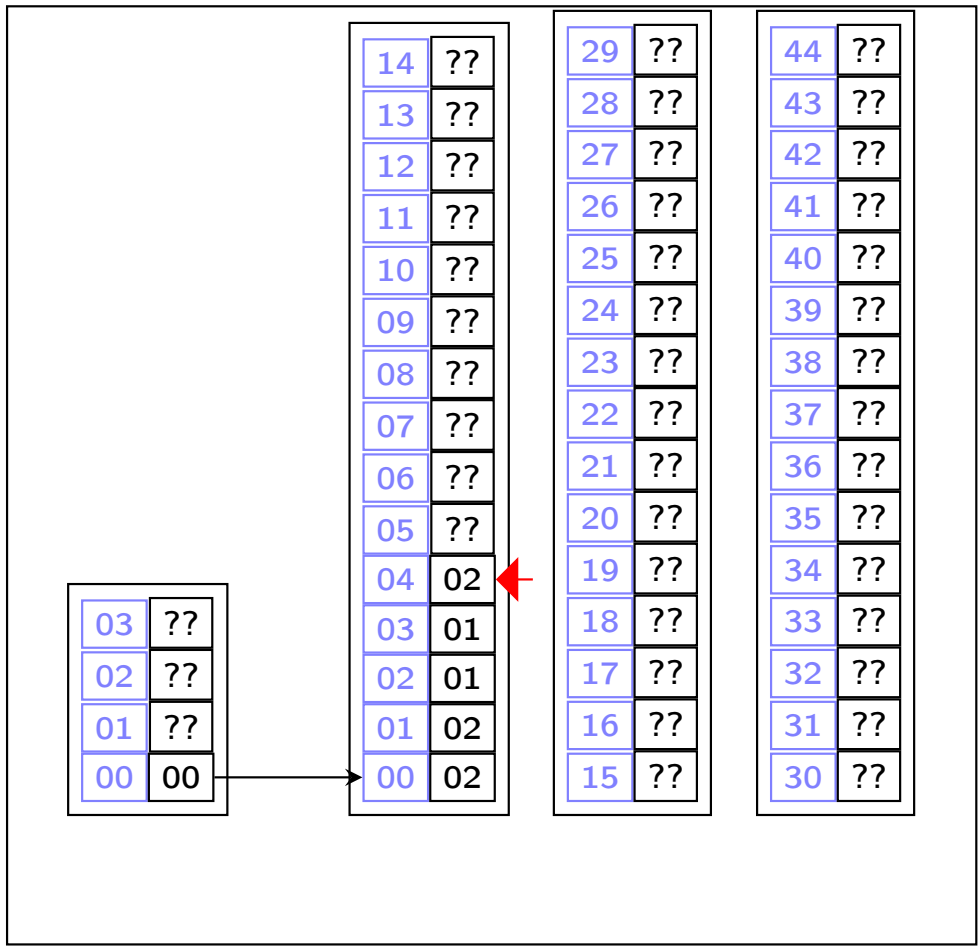
Segunda Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



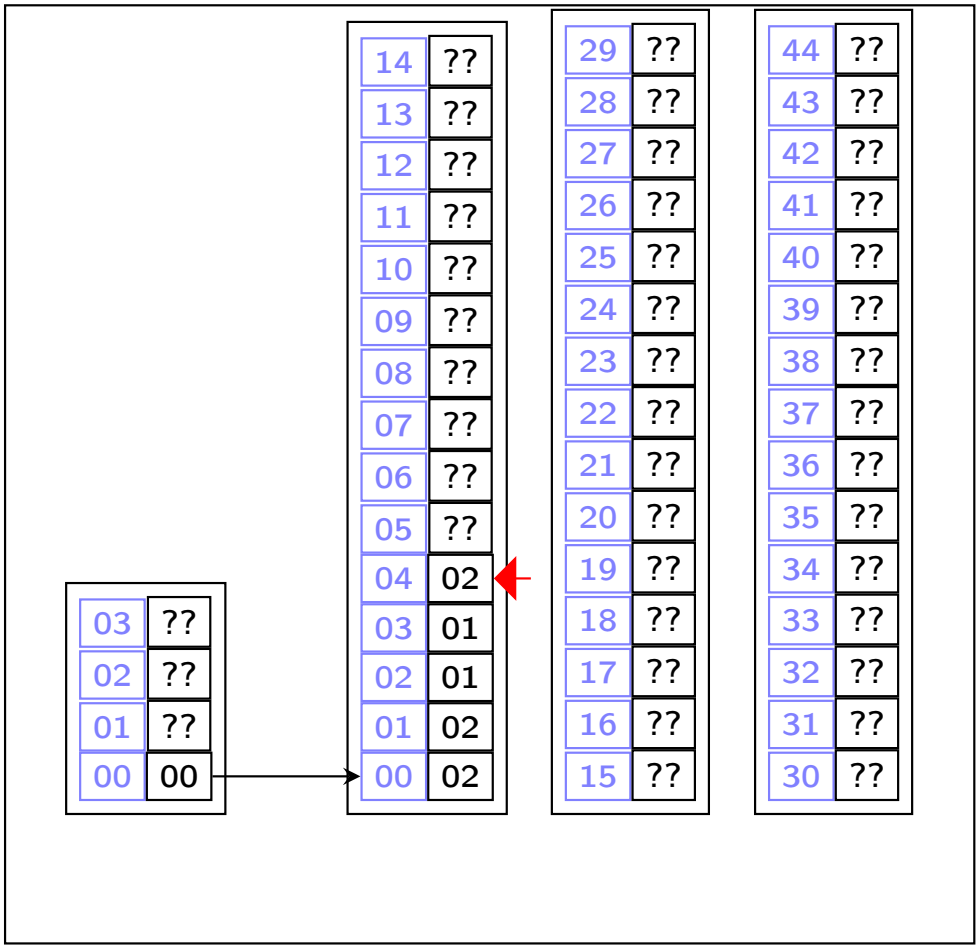
Segunda Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



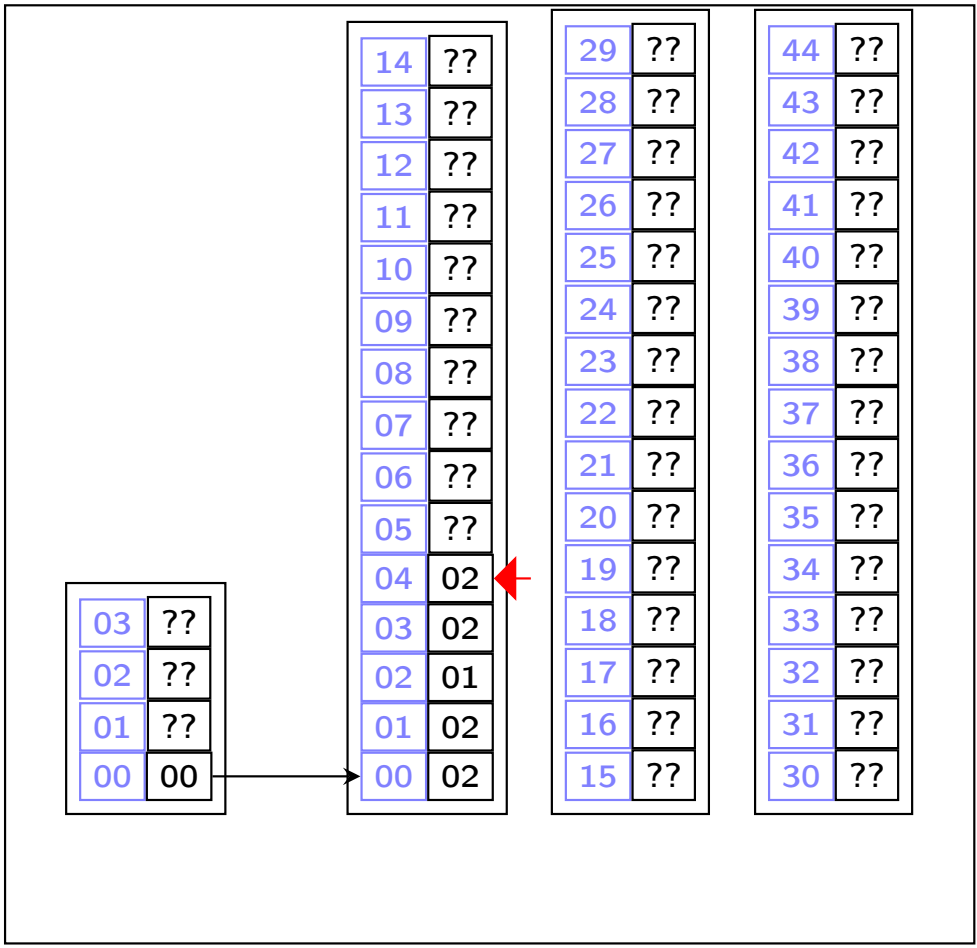
Segunda Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



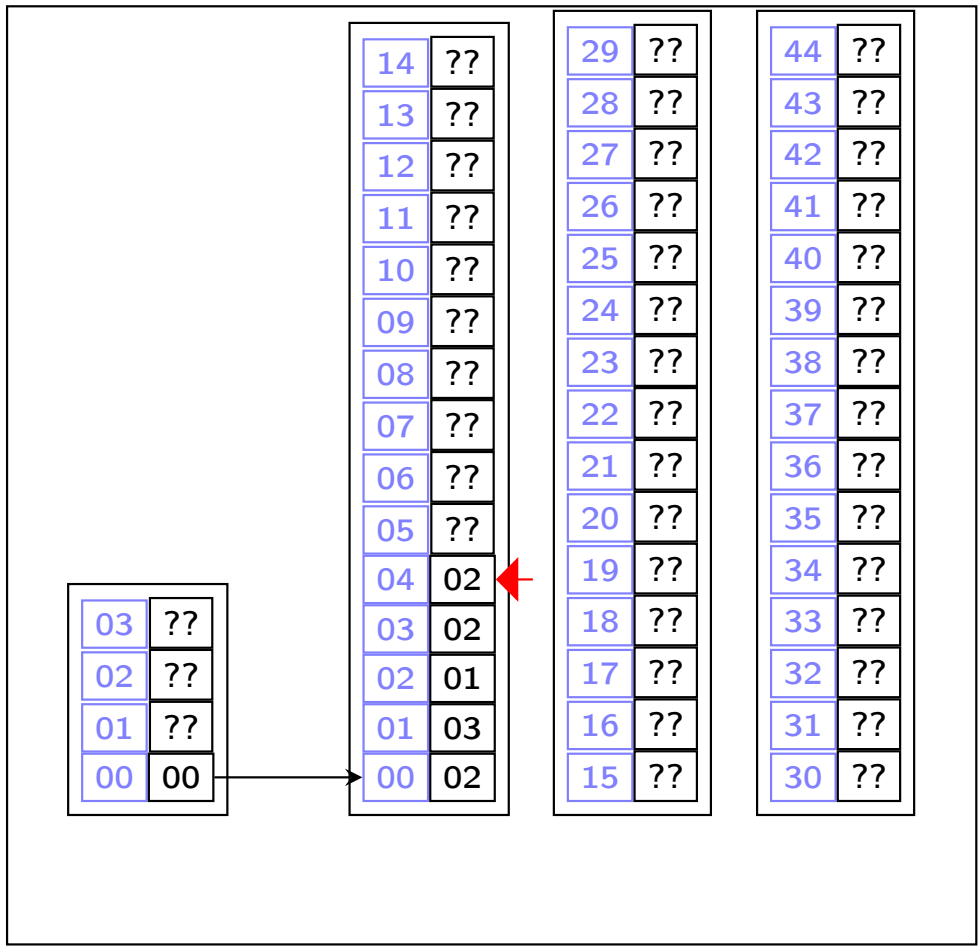
Segunda Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



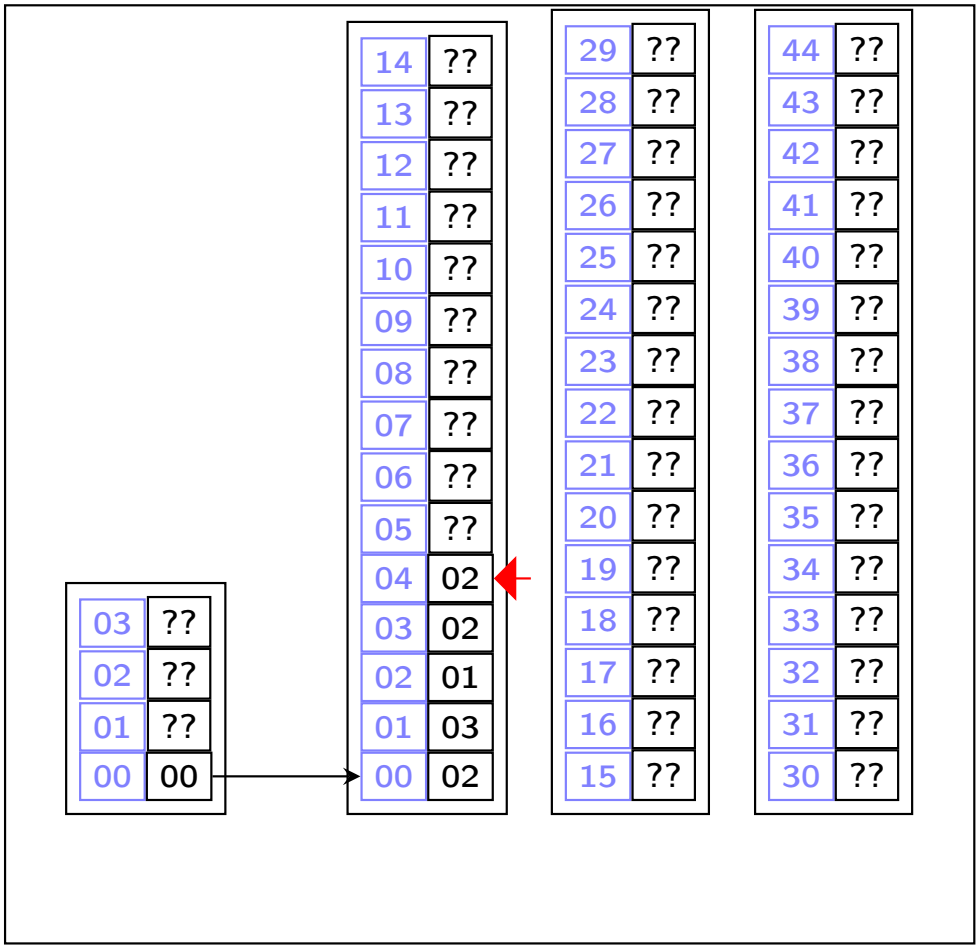
Segunda Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:  NADA            CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



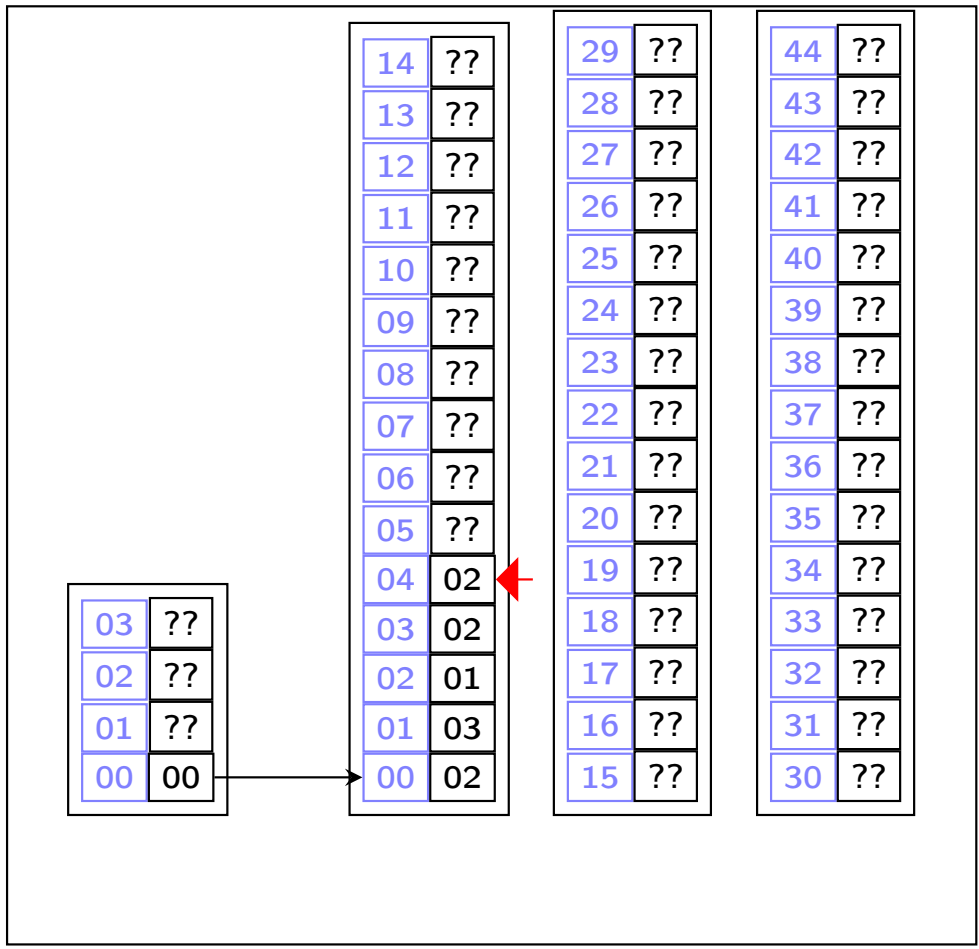
Segunda Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



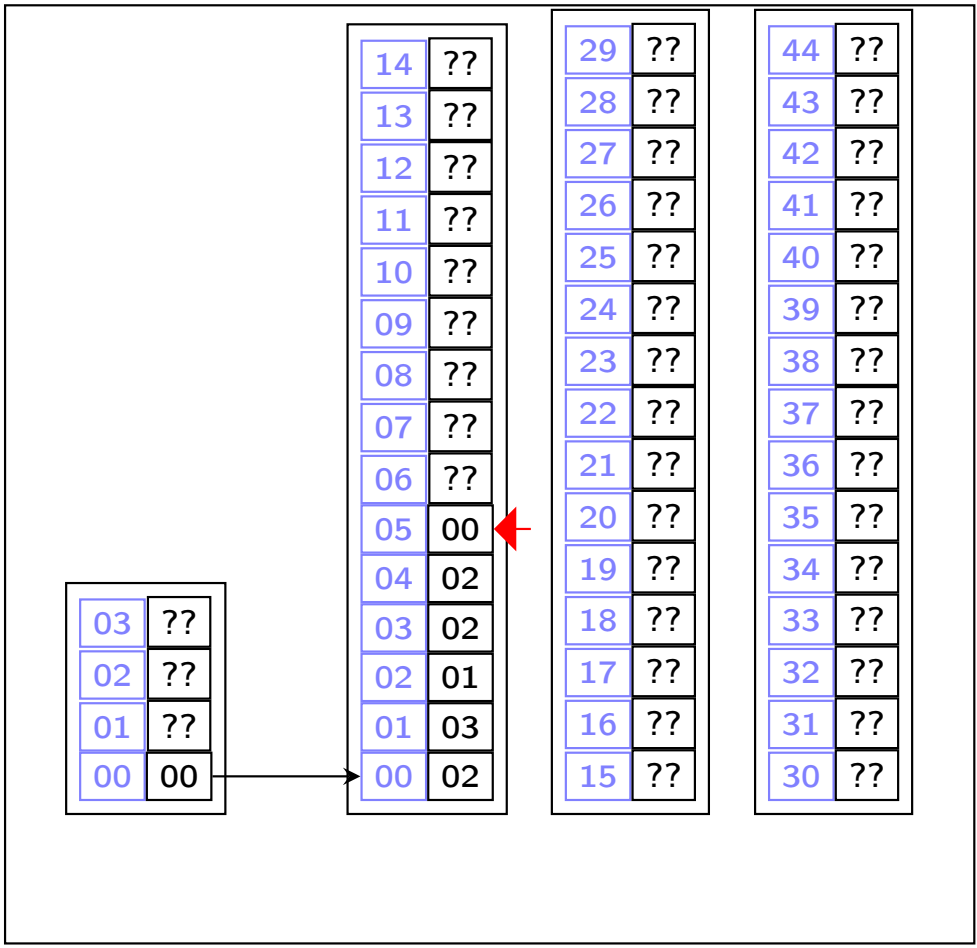
Terceira Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:  NADA            CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



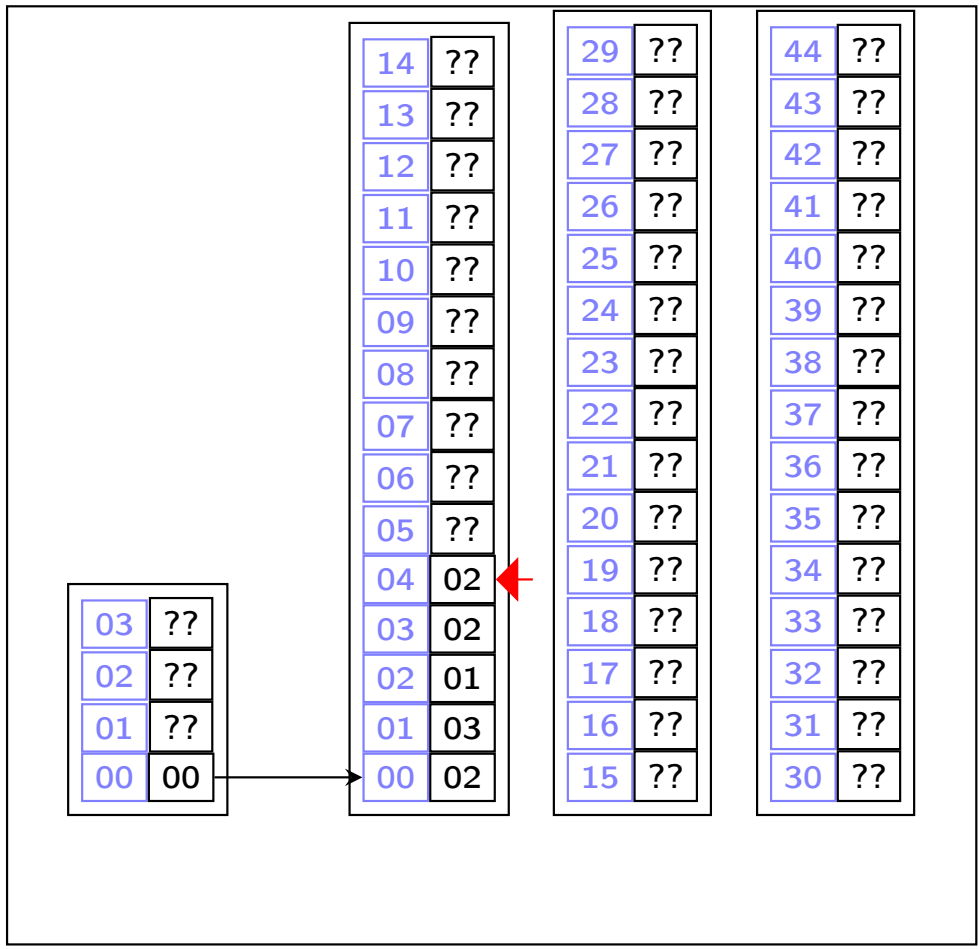
Terceira Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



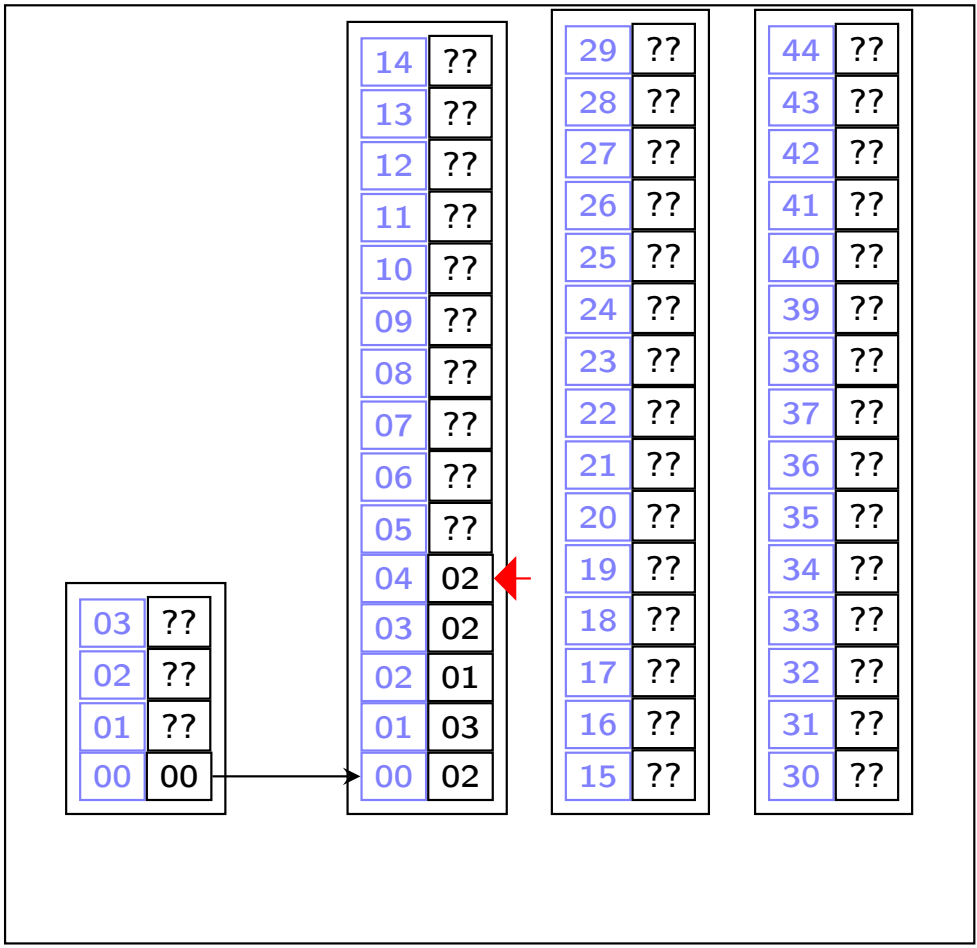
Terceira Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



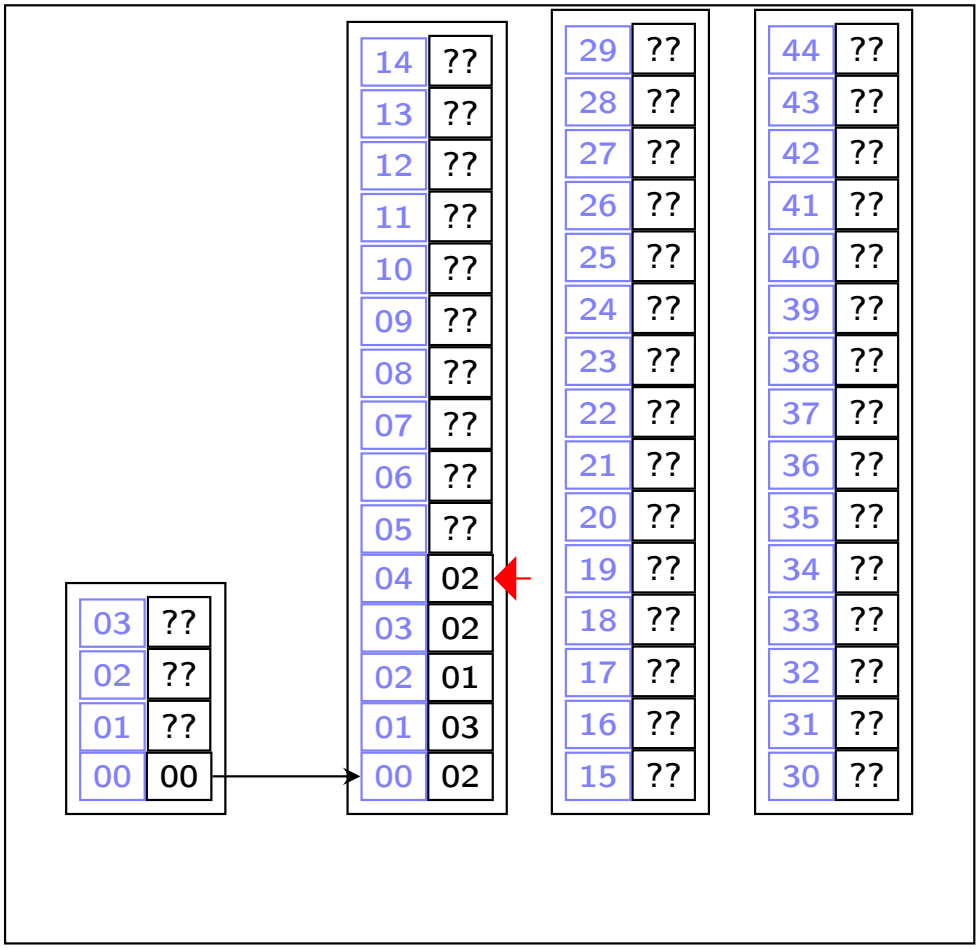
Terceira Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:  NADA            CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



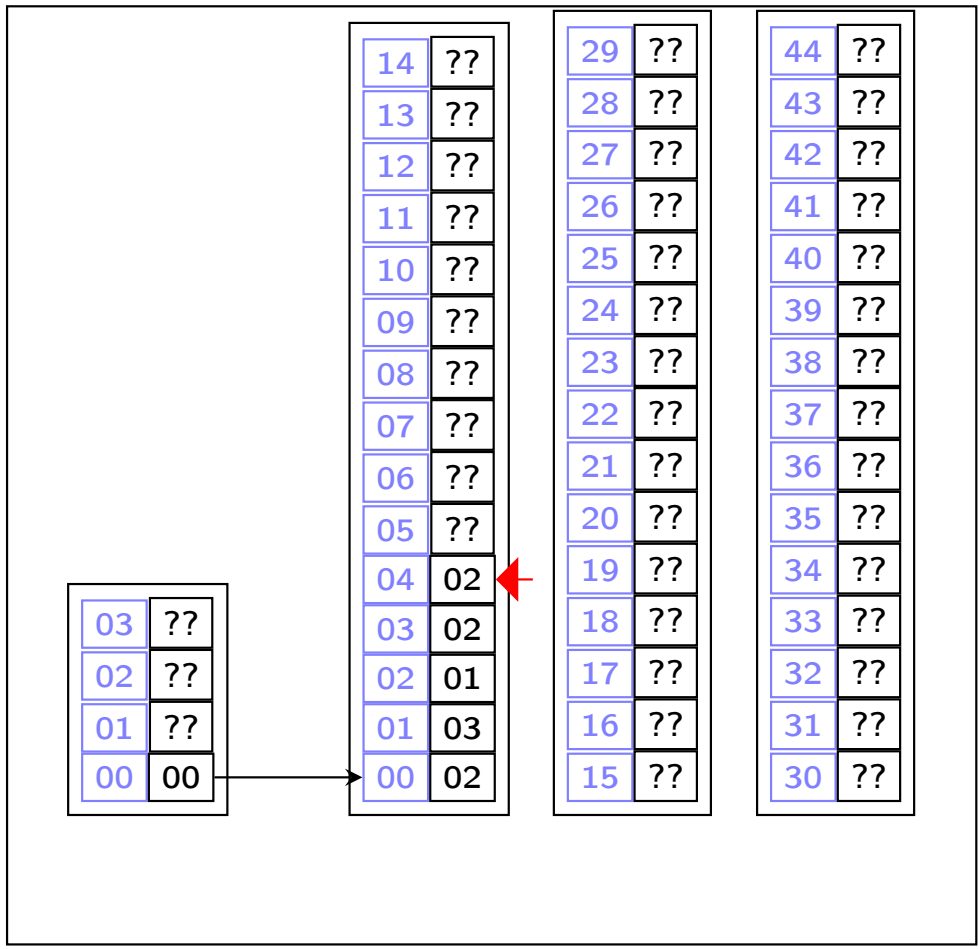
Terceira Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



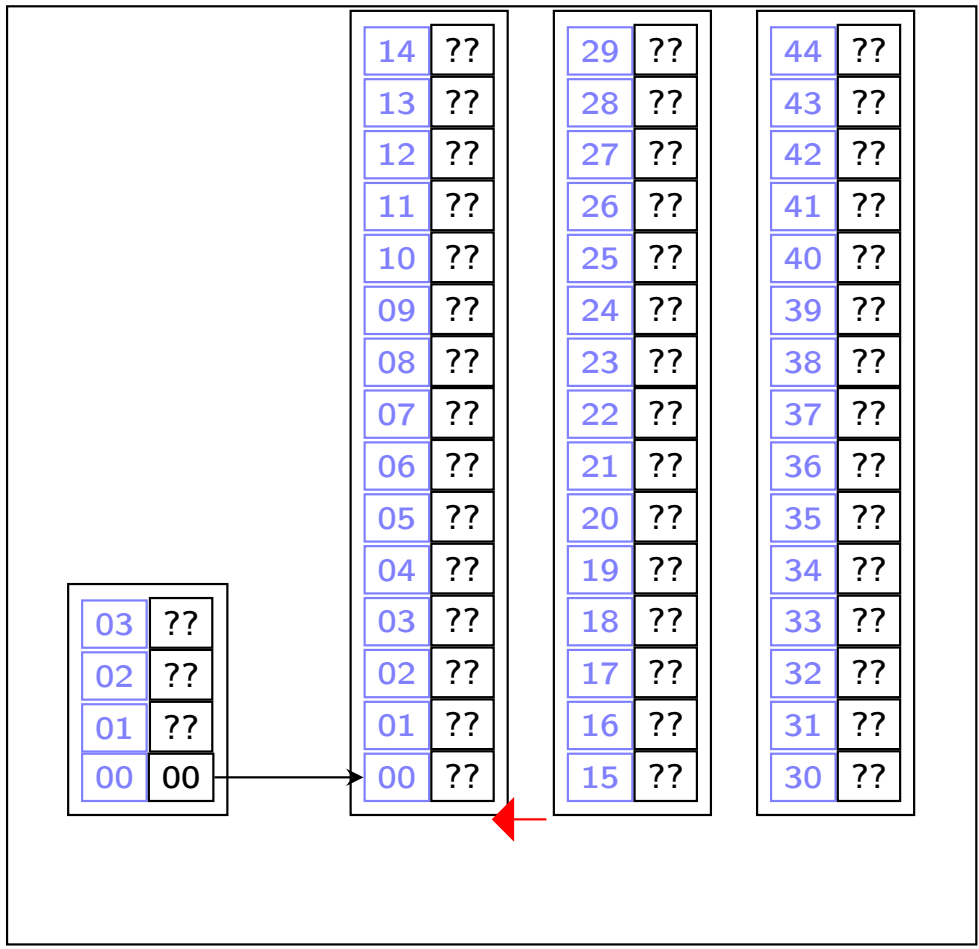
Terceira Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



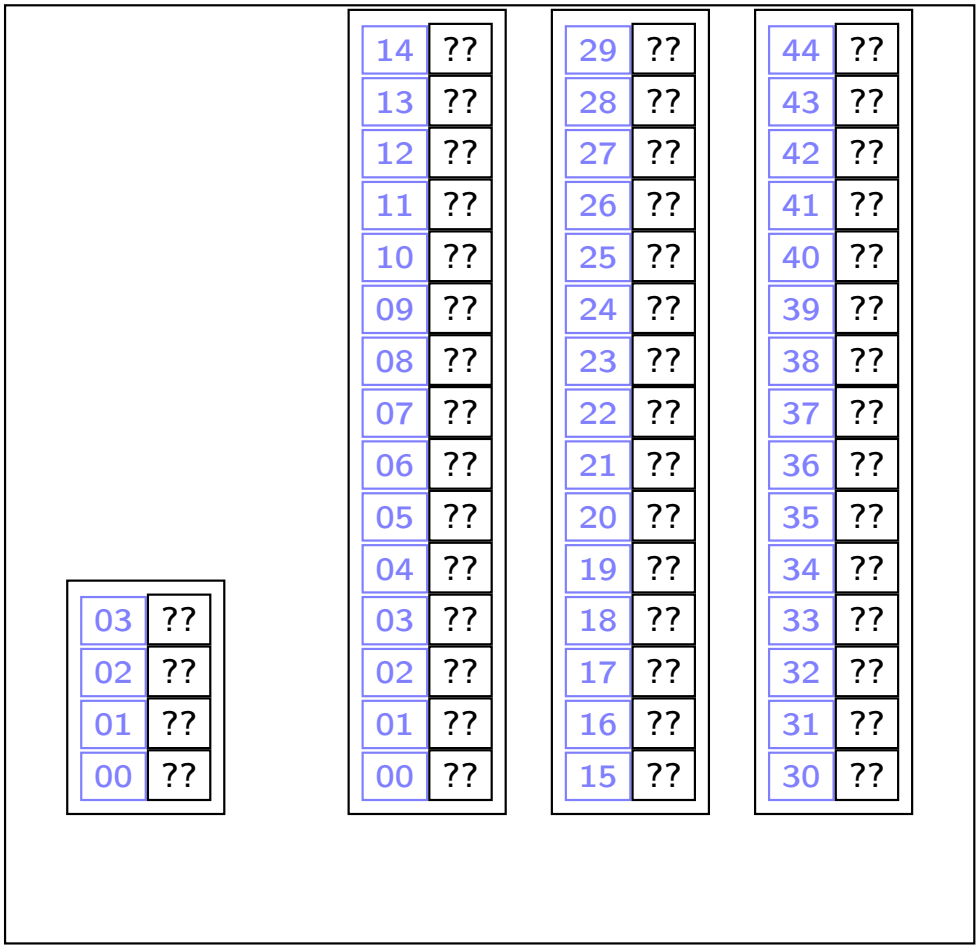
Terceira Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:  NADA            CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```



Terceira Iteração

```
      INPP            CRVL 0,3
      AMEM 2          ARMZ 0,2
      AMEM 3
      LEIT            CRVL 0,4
      ARMZ 0,0        ARMZ 0,3
      CRCT 0
      ARMZ 0,2        CRVL 0,1
      CRCT 1          CRCT 1
      ARMZ 0,3        ARMZ 0,1
      CRCT 1          DSVS R00
      ARMZ 0,1  R01:  NADA
R00:NADA              CRVL 0,0
      CRVL 0,1        IMPR
      CRVL 0,0        CRVL 0,1
      CMEG            IMPR
      DSVF R01        DMEM 5
      CRVL 0,3        PARA
      CRVL 0,2
      SOMA
      ARMZ 0,4
```

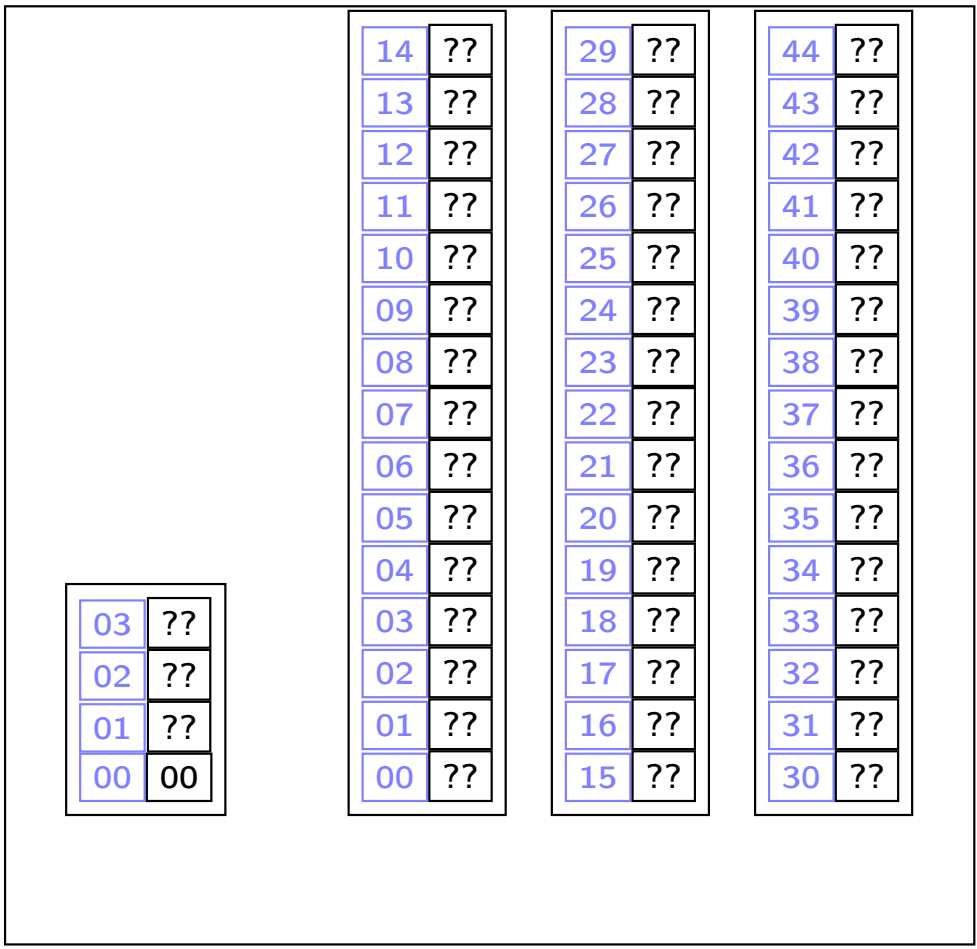


Exec.Pascal

- A execução das instruções MEPA uma a uma apresenta vários problemas didáticos:
  - É muito detalhada, podendo causar erros.
  - O nível de detalhe esconde muitas informações pertinentes, como por exemplo: Qual é a variável que reside no endereço léxico 0,0?
- Por estas e outras razões fiz a opção de simular a execução do programa Pascal diretamente na máquina virtual.
- Isto aumenta o nível de abstração e permite explicar o funcionamento conceitual do ambiente de execução.
- Se existe alguma coisa positiva com a execução anterior, acredito que seja saber como se sente um computador.

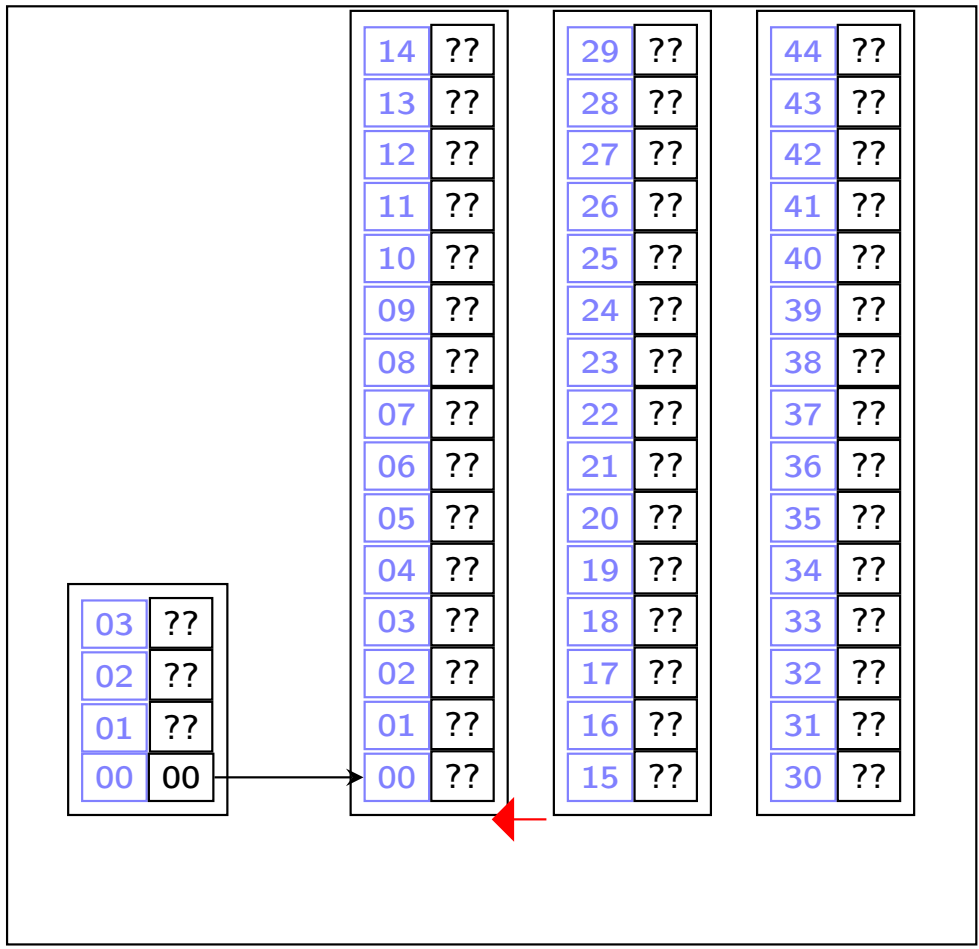
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```



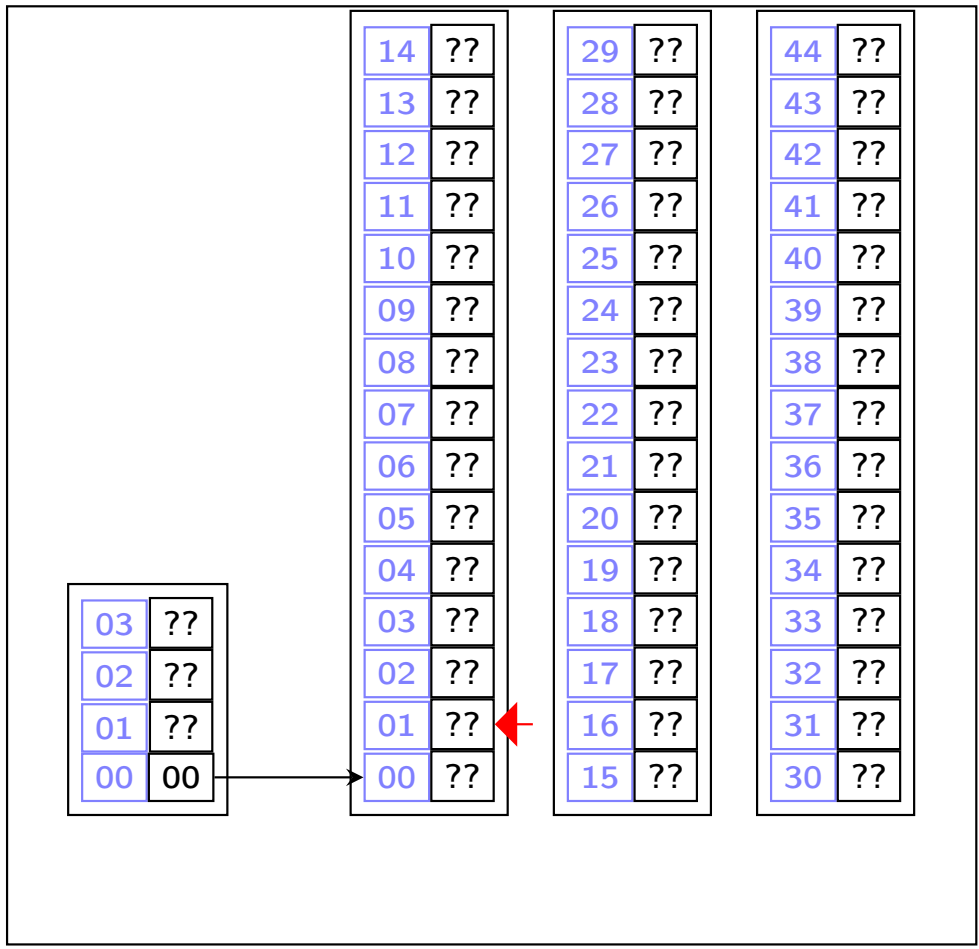
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```



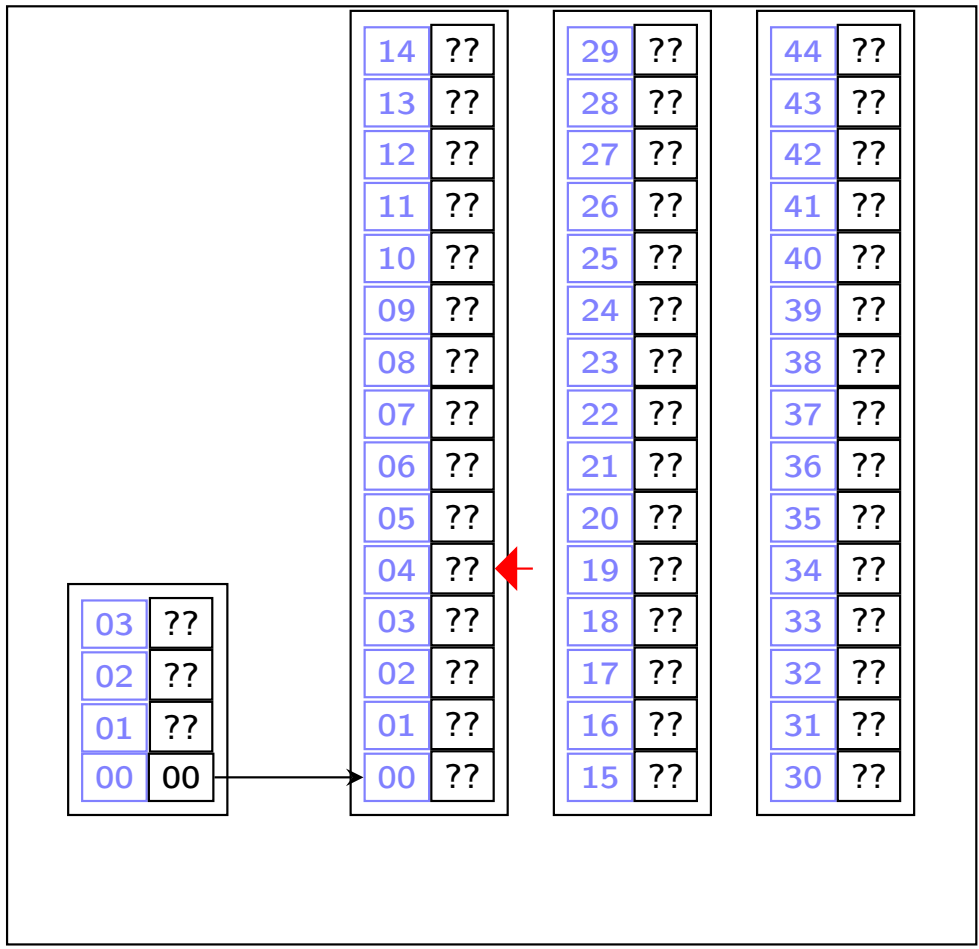
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```



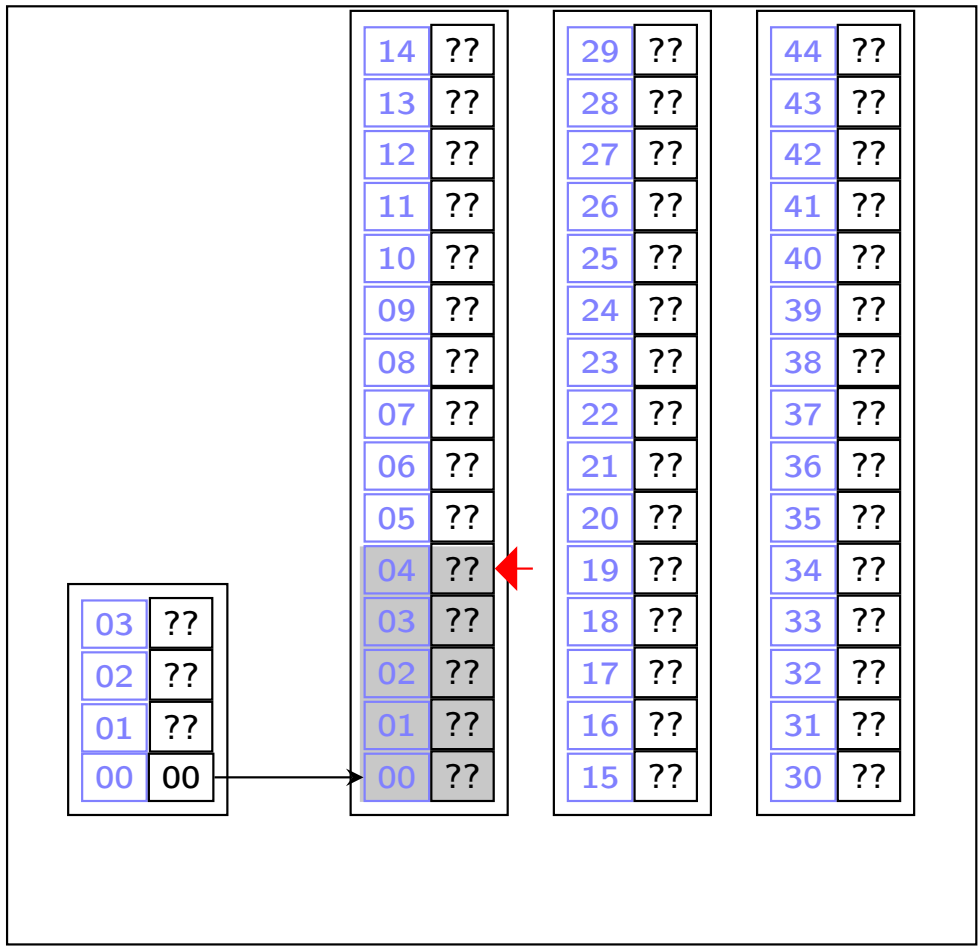
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```



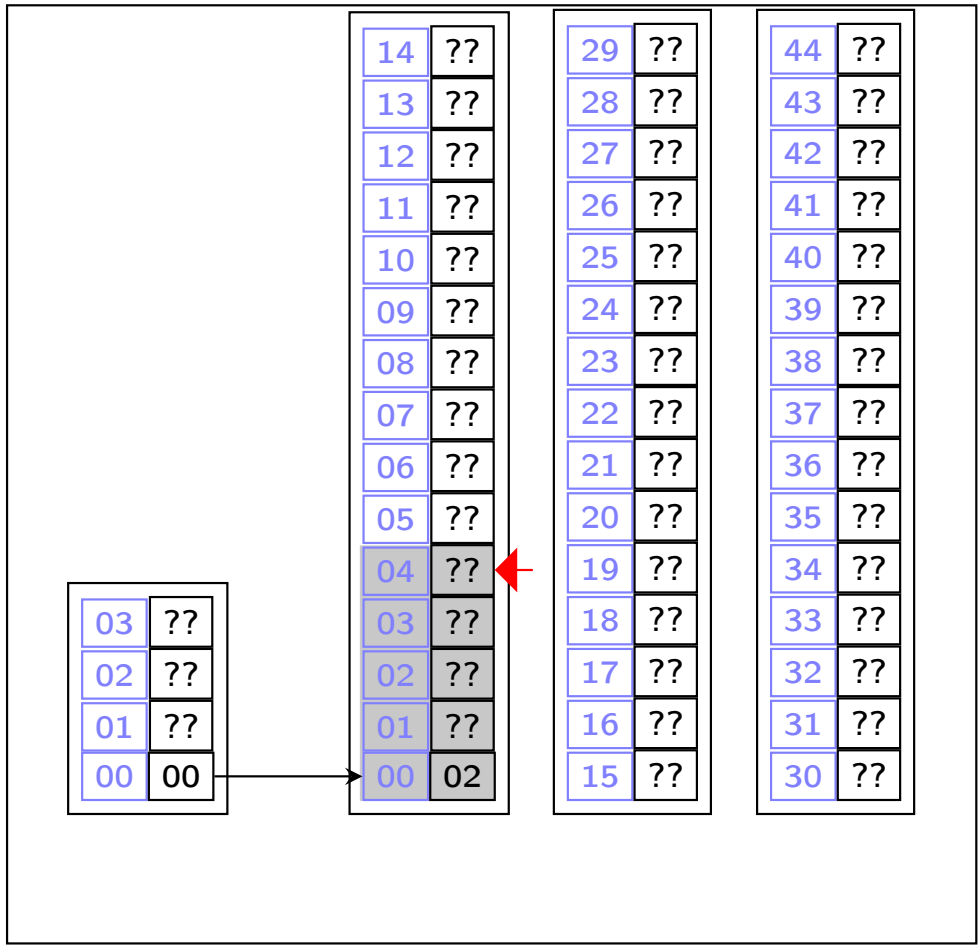
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



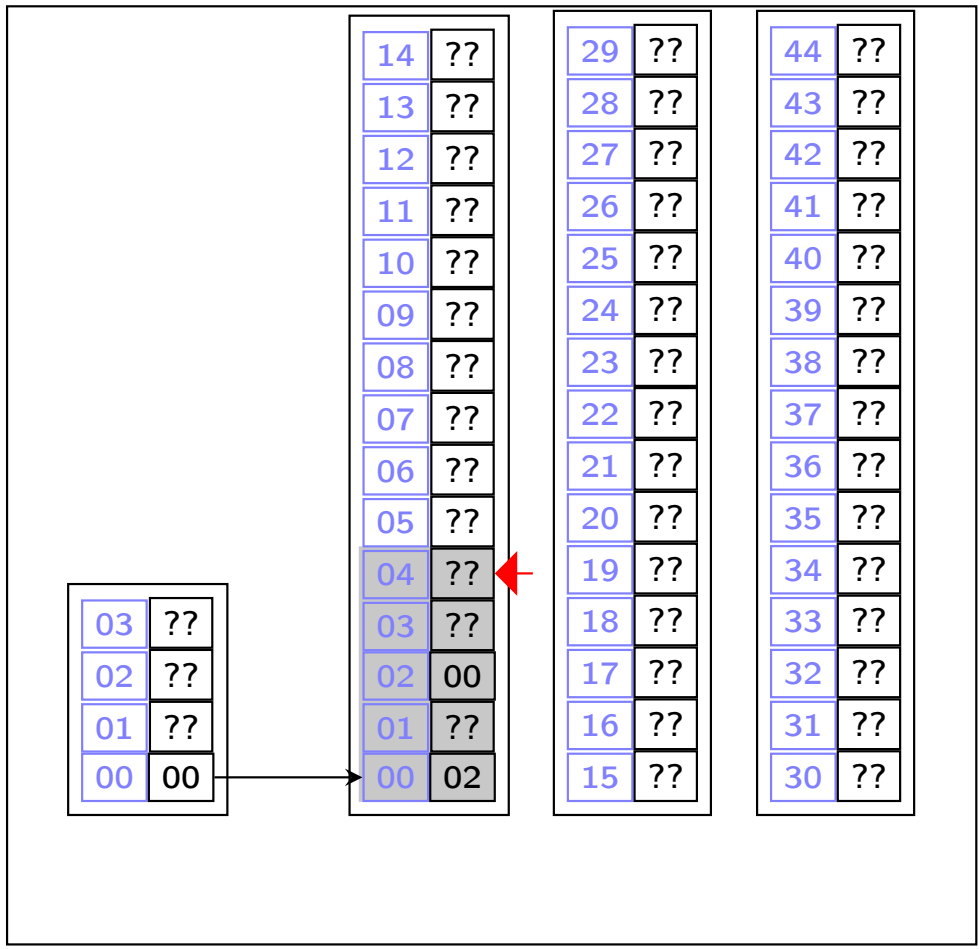
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



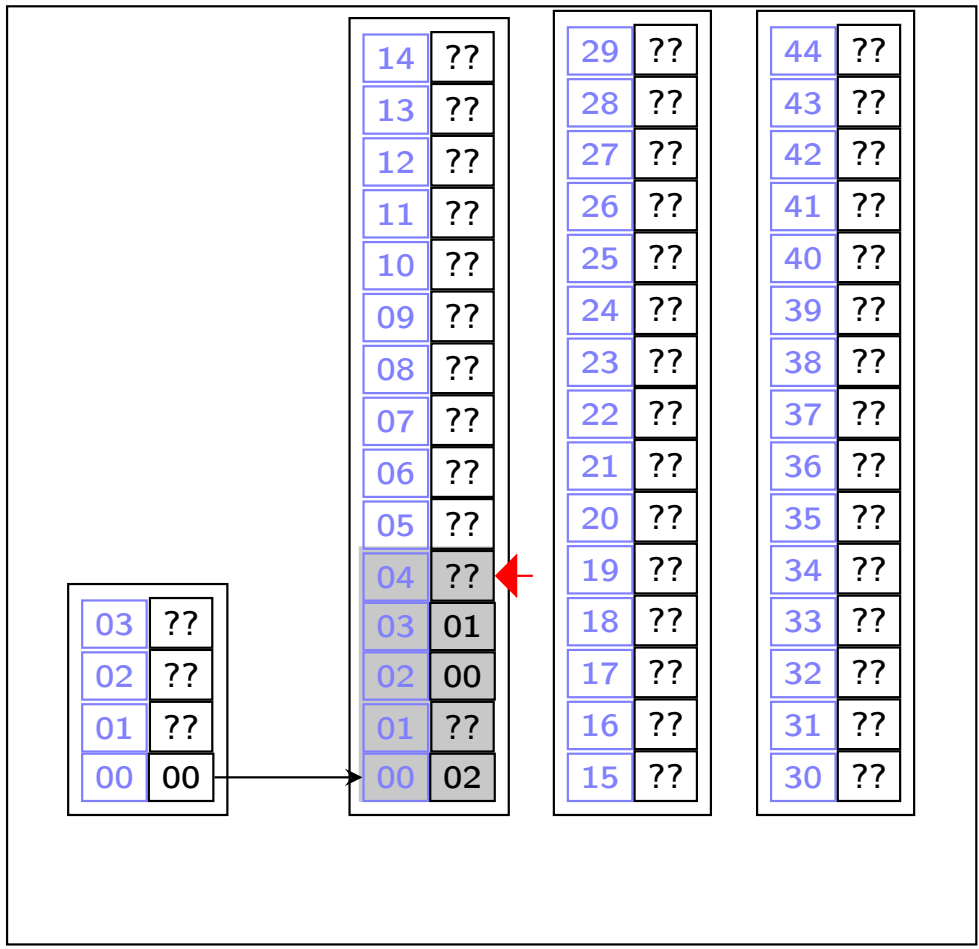
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



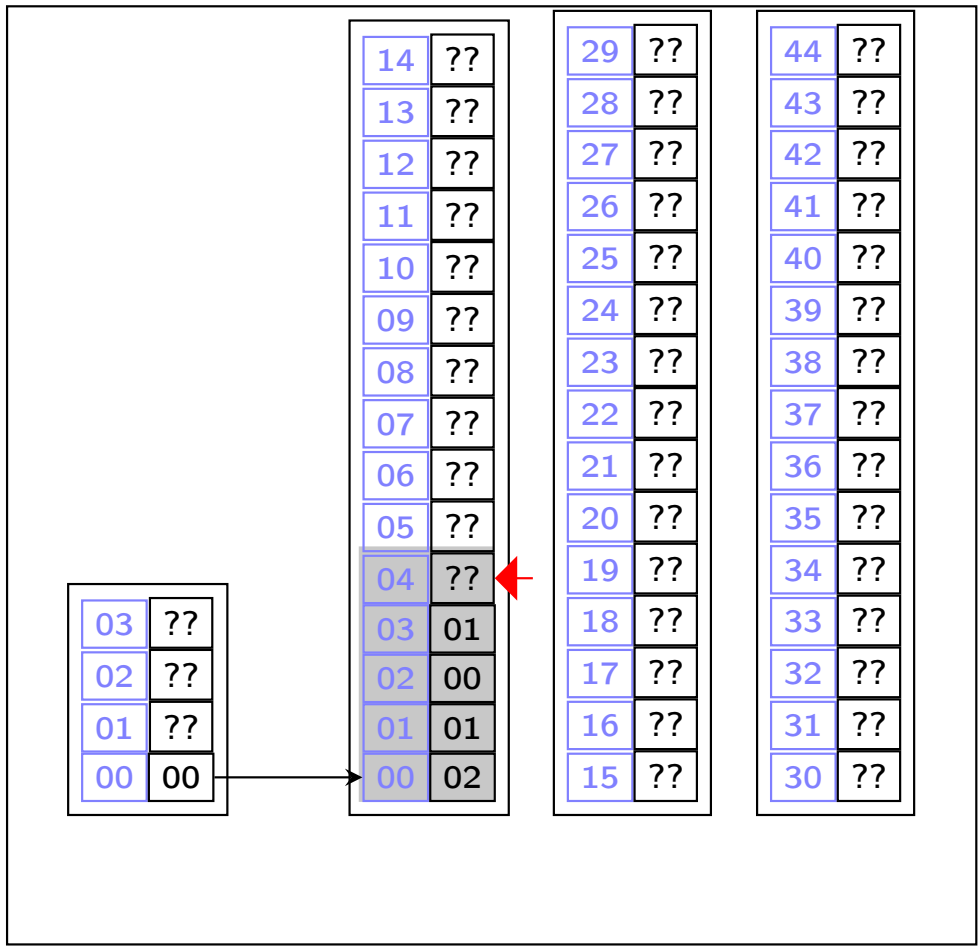
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



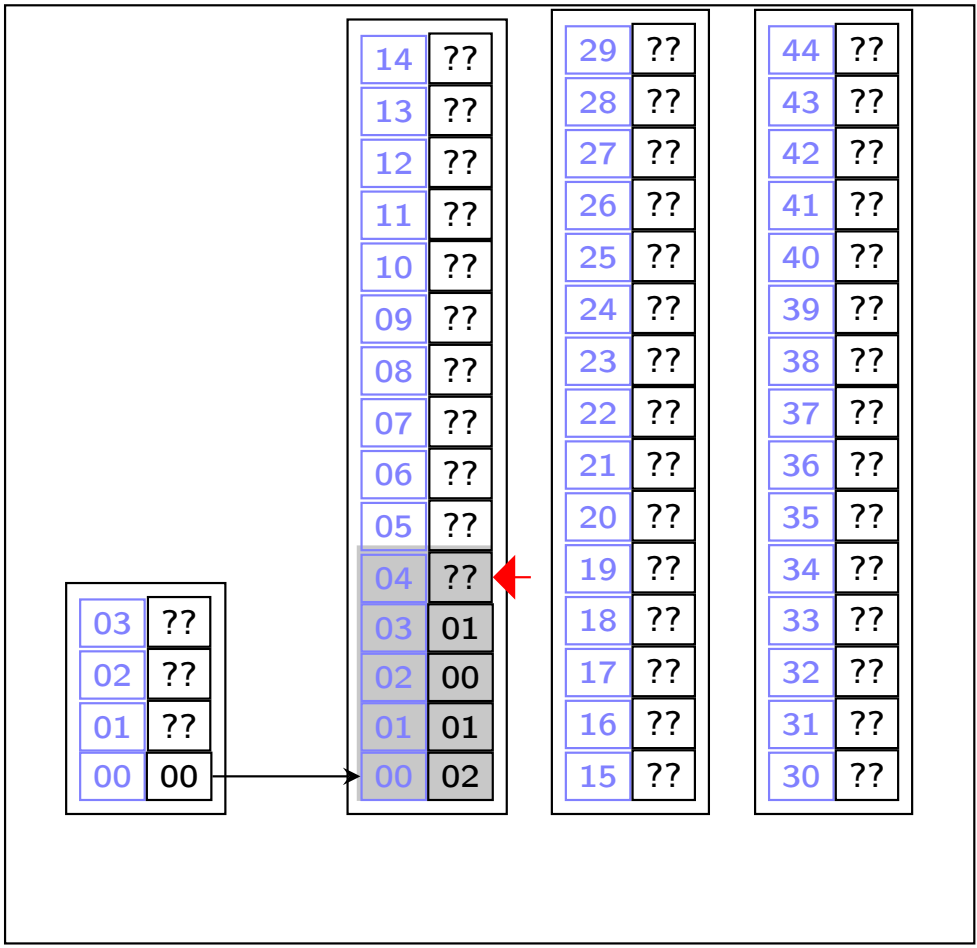
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



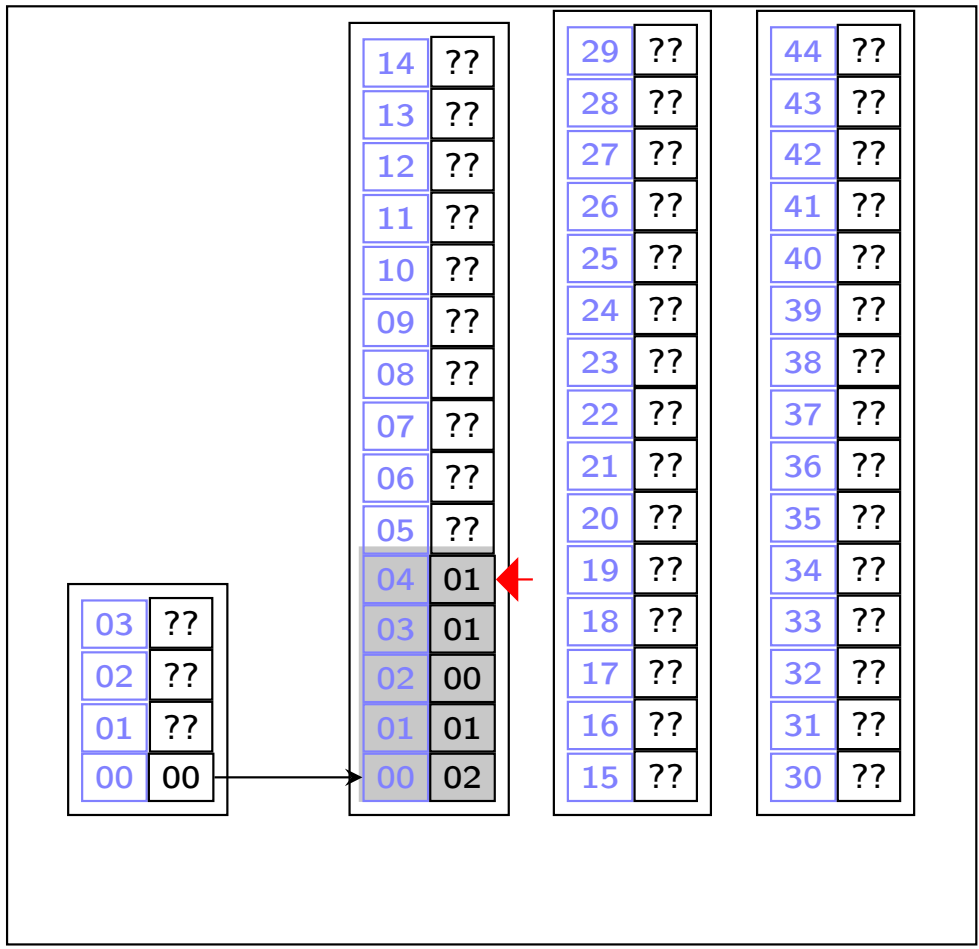
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



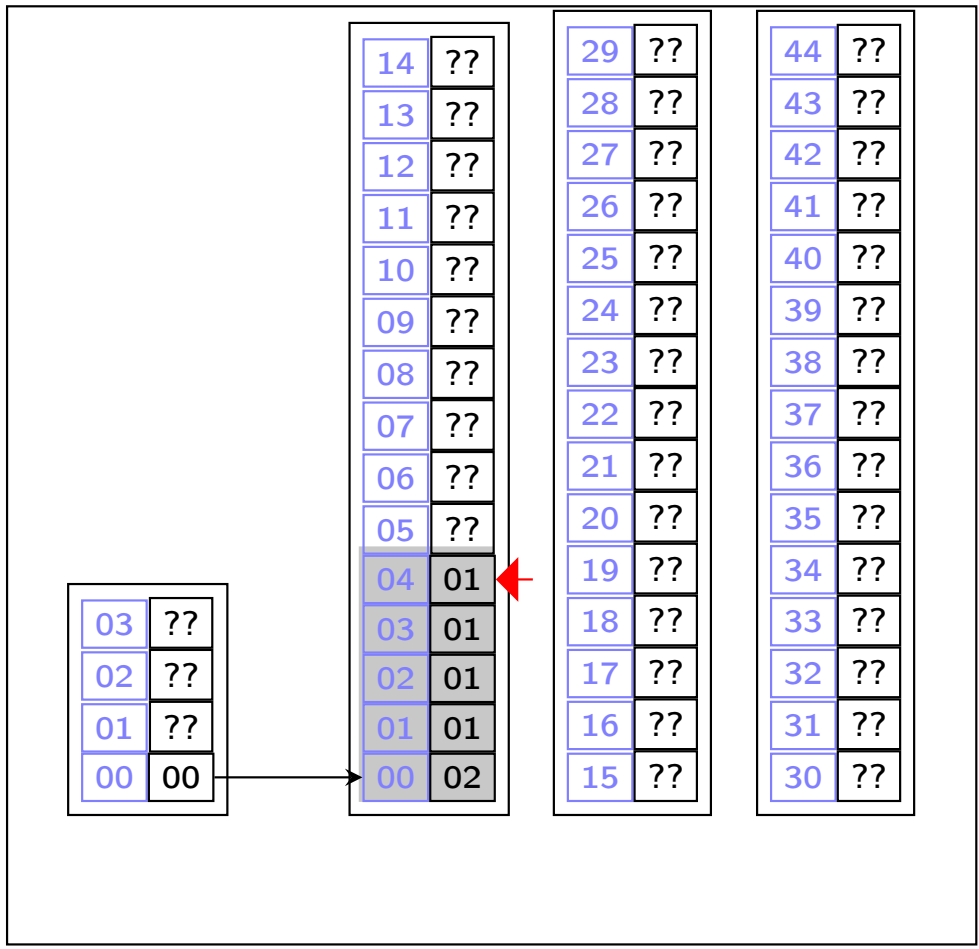
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



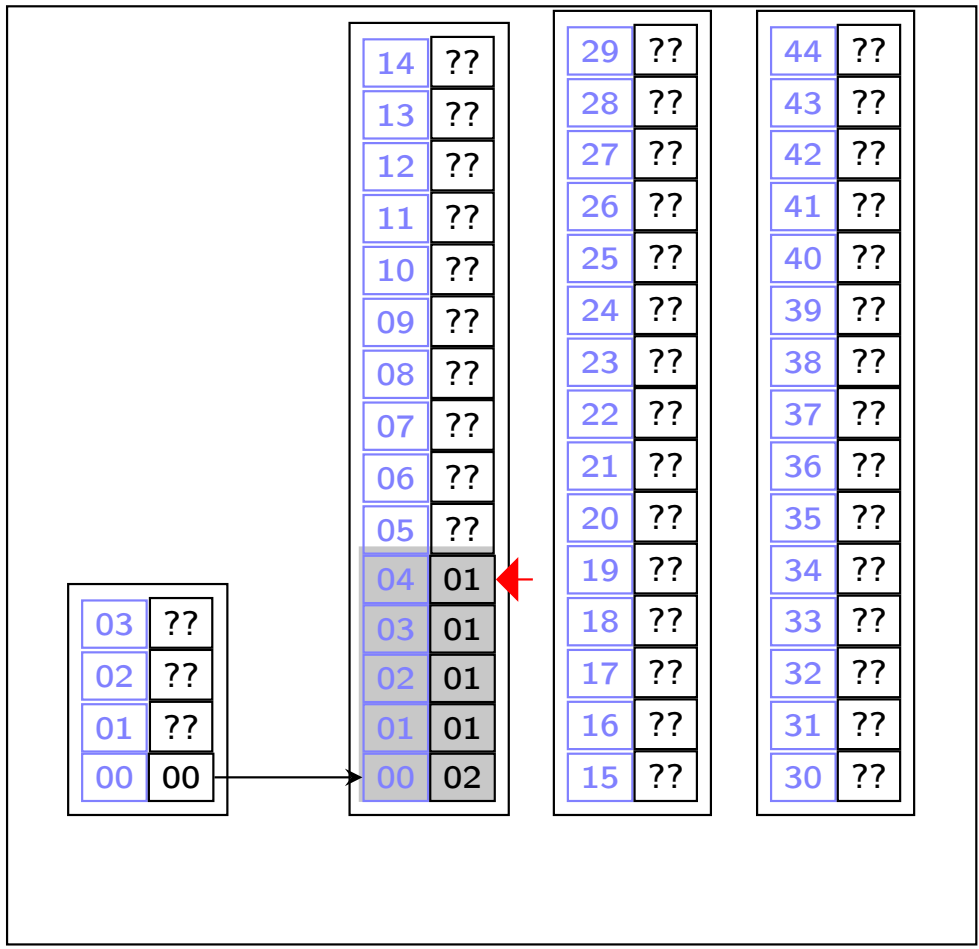
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



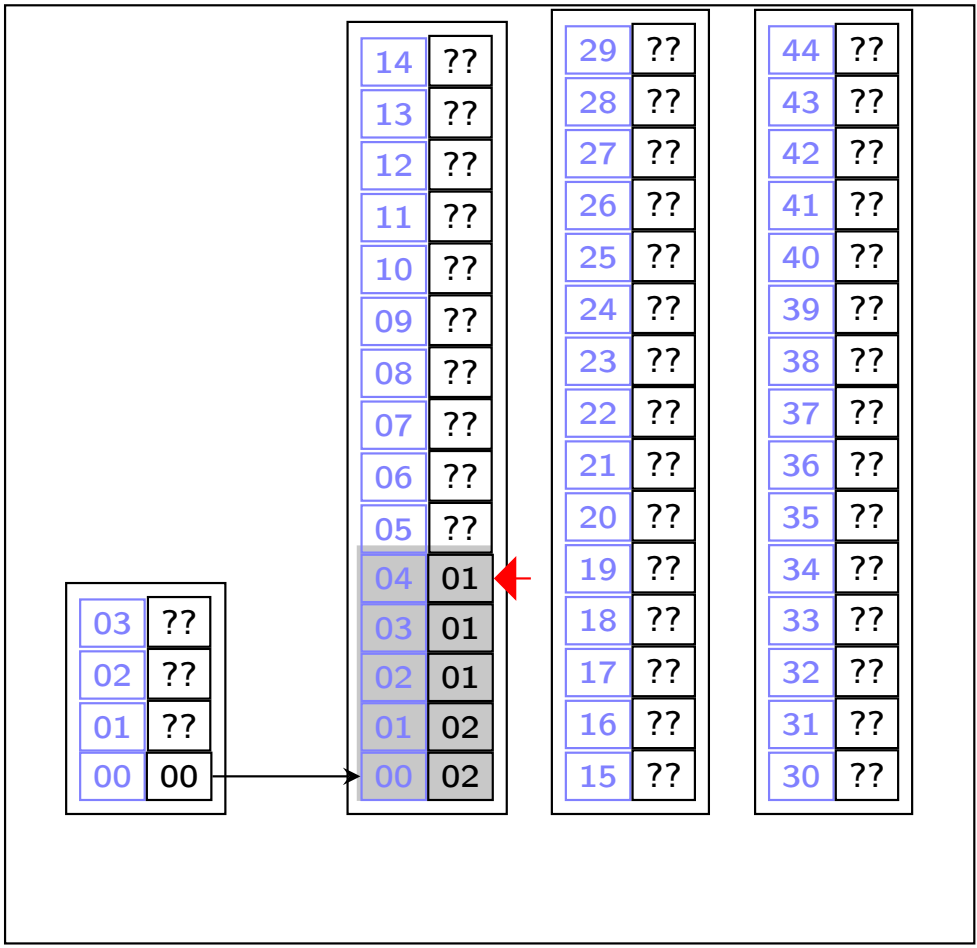
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



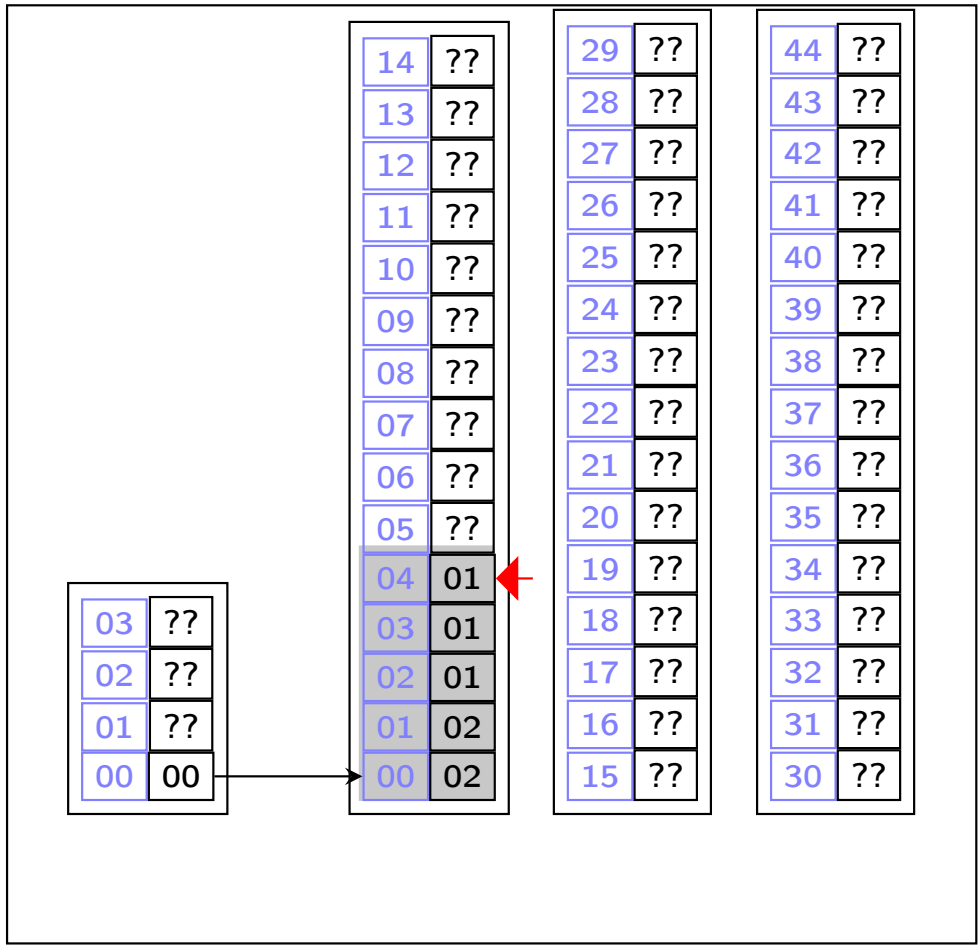
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



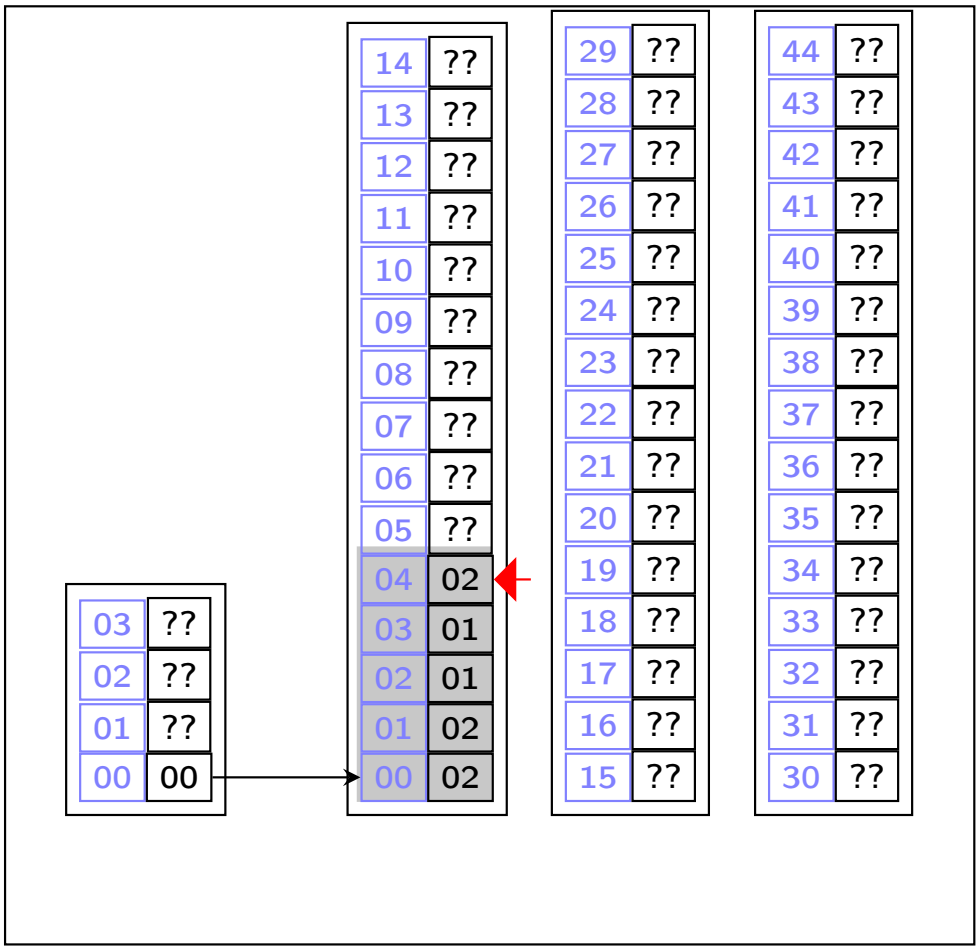
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



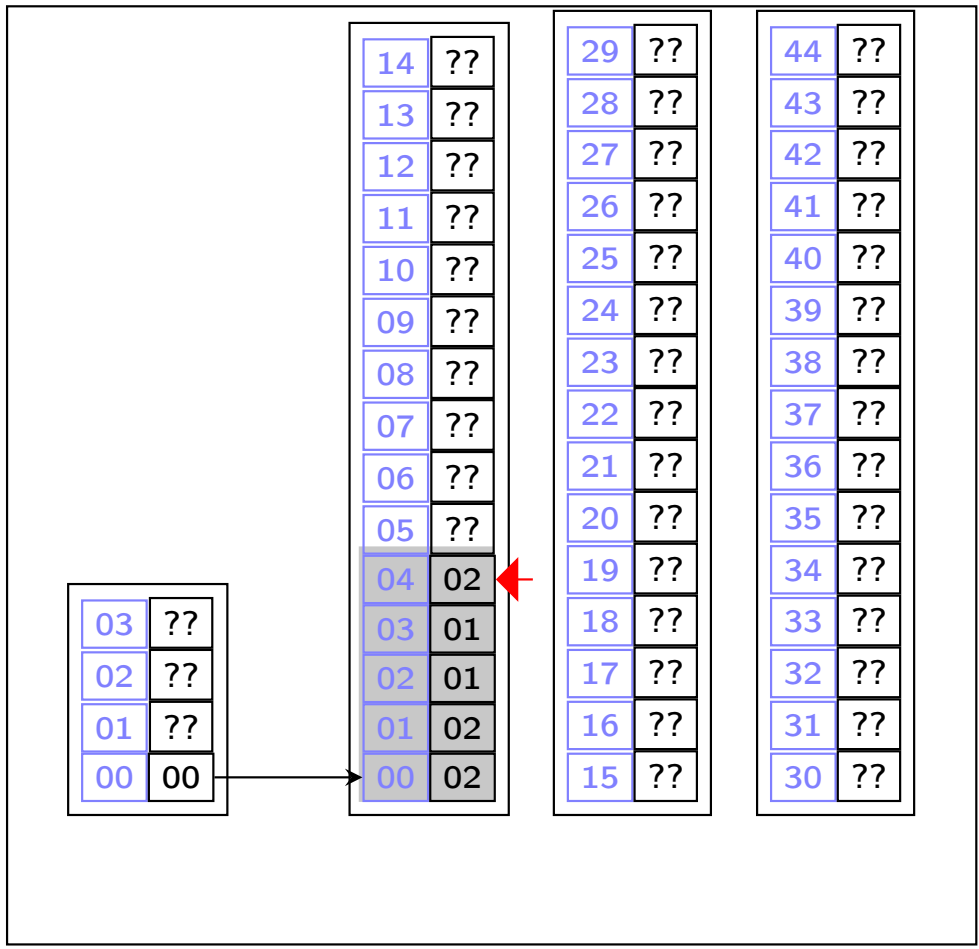
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



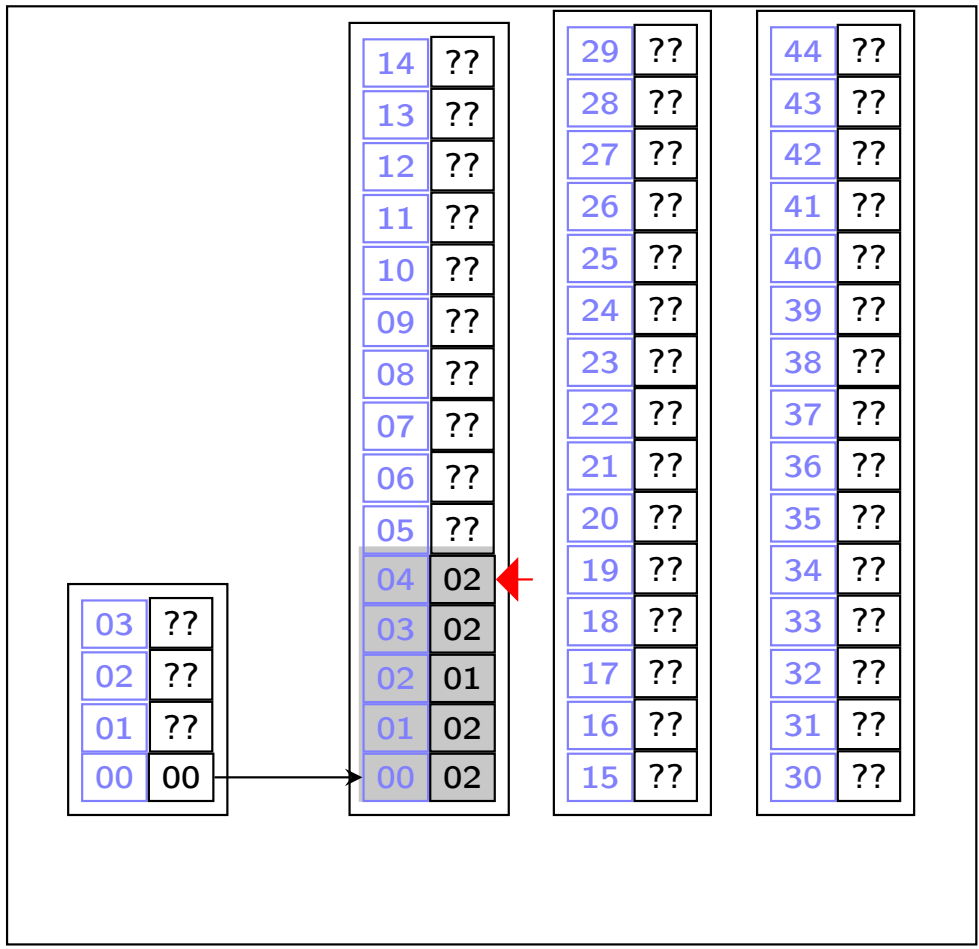
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



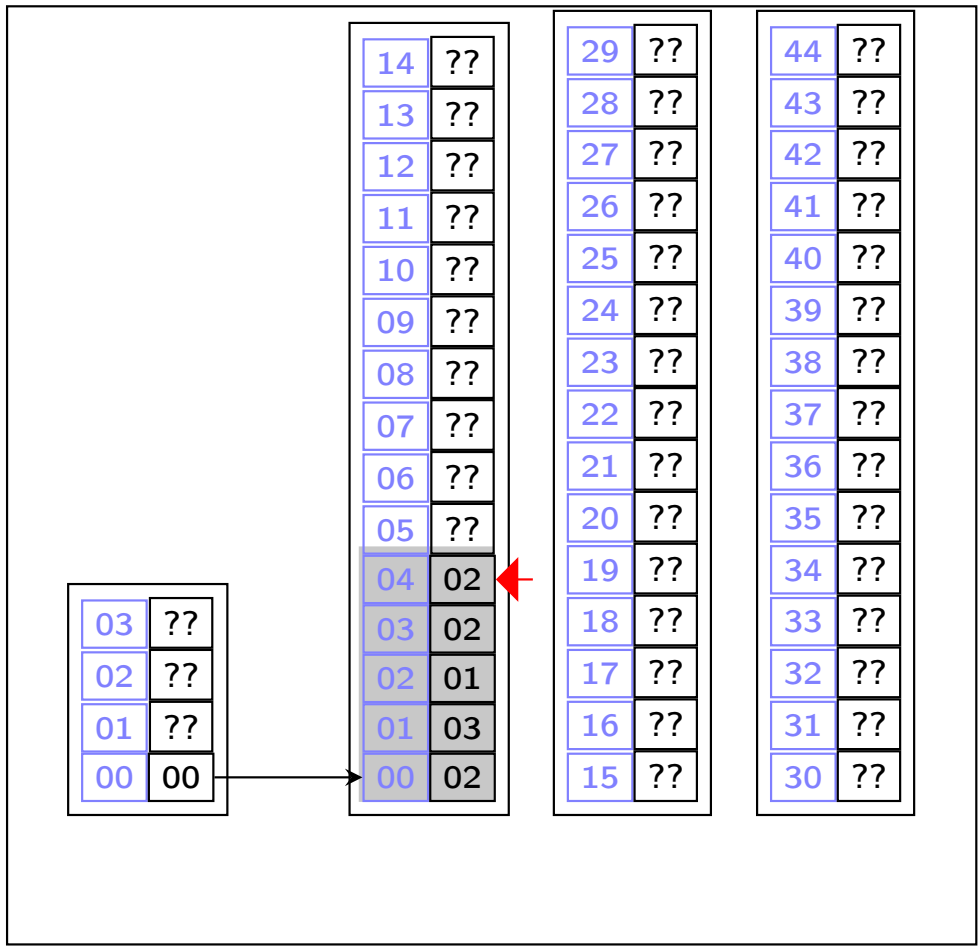
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



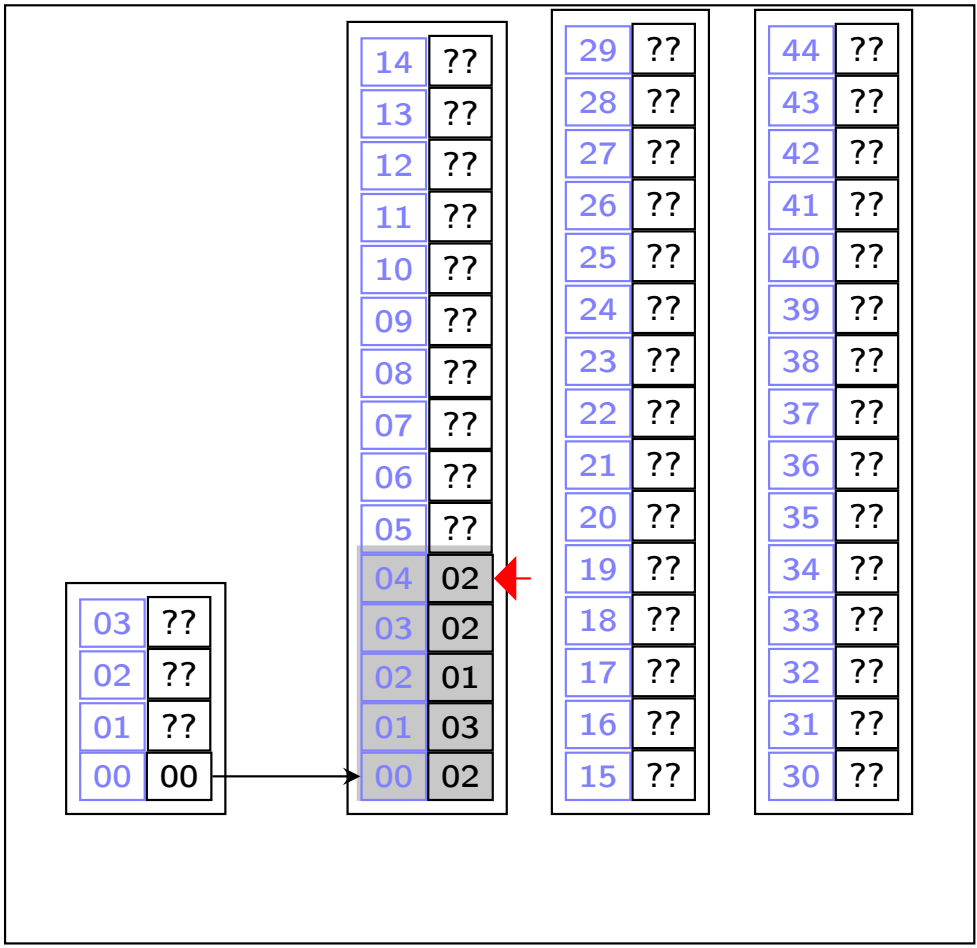
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



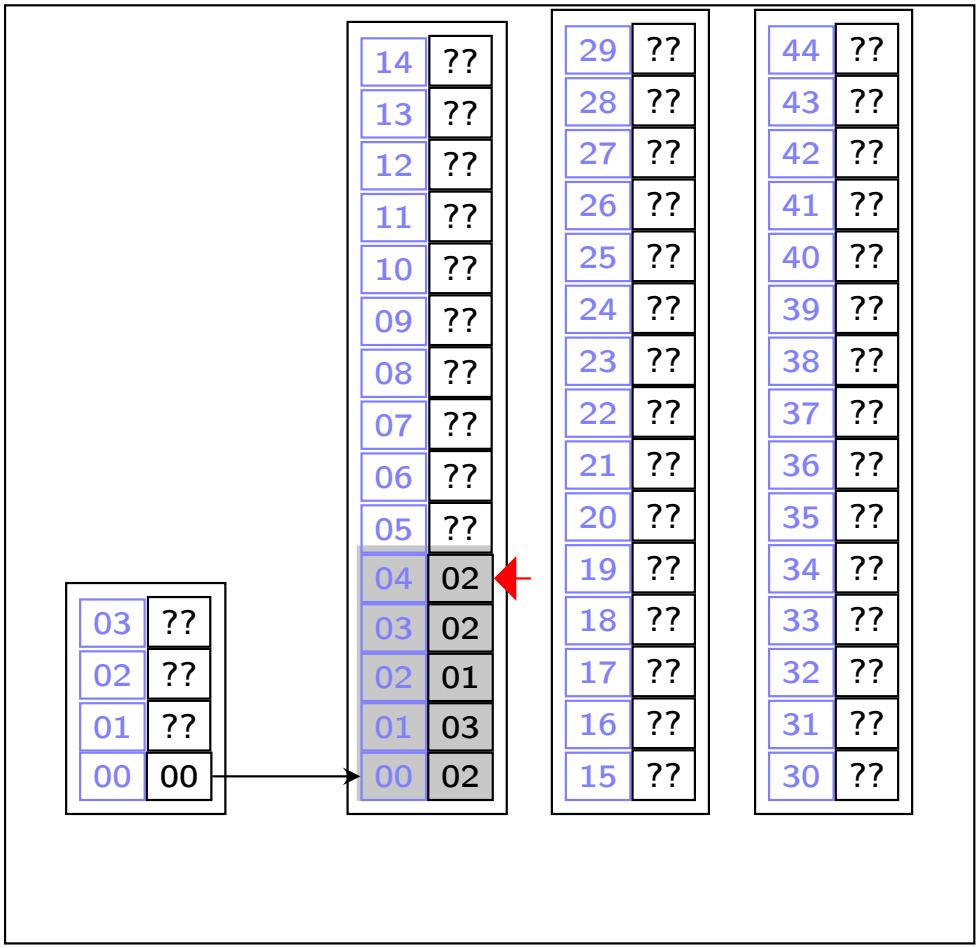
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



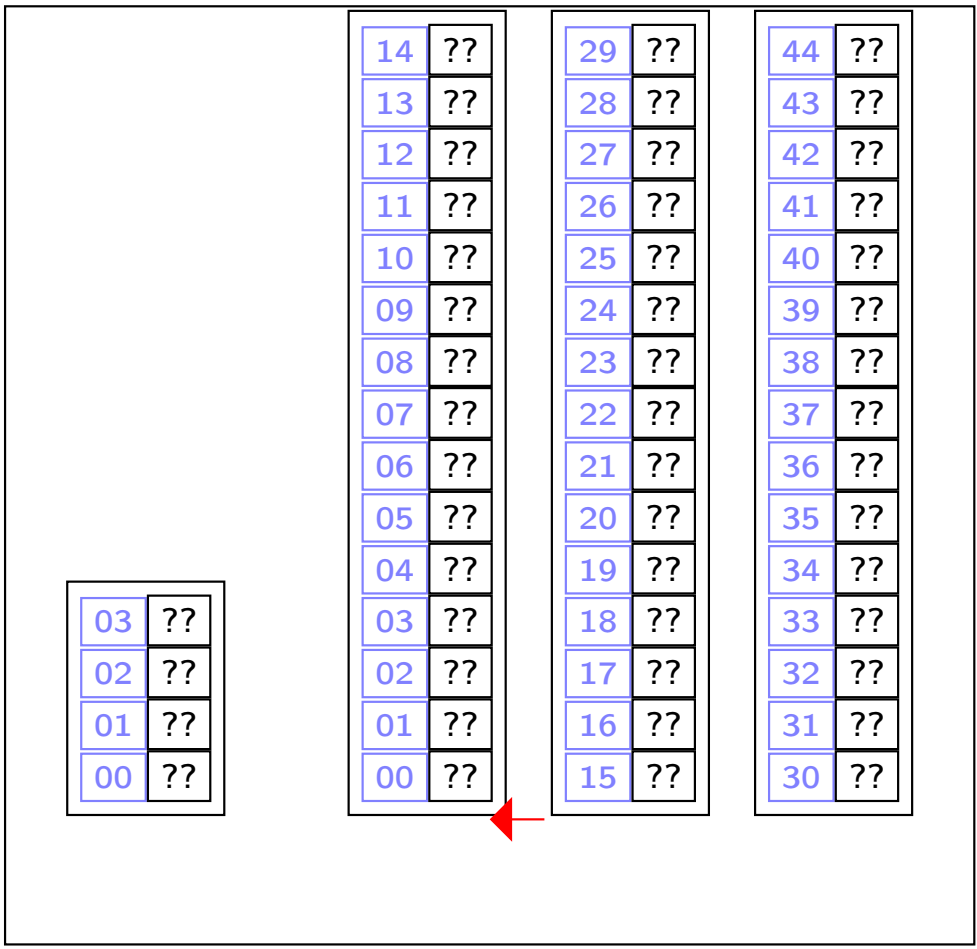
Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.

f3(0,4)
f2(0,3)
f1(0,2)
k (0,1)
n (0,0)
```



Execução

```
program comandoWhile;
var n, k: integer;
    f1, f2, f3:integer;
begin
   read(n)
   f1:=0; f2:=1; k:=1;
   while (k <= n) do
   begin
      f3:=f2+f1;
      f1:=f2;
      f2:=f3;
      k:=k+1
   end;
   write(n,k)
   end.
```



Projeto

- Implemente o comando `while` no compilador.
  - Baseie-se no esquema de tradução.
  - Pense em como contornar o problema de vários comandos `while` aninhados. Os diversos rótulos de entrada (`R00`) e de saída (`R01`) não devem ser confundidos.
  - Sugestões:
    - Criar uma subrotina "geradora" do próximo rótulo.
    - Criar uma pilha de rótulos.
    - No início do `while`, empilhar os dois rótulos.
    - Ao final, desempilhá-los.

- Página para anotações

Licença

- Slides desenvolvidos somente com software livre:
  - LaTeX usando beamer;
  - Inkscape.
- Licença:
  - Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License. `http://creativecommons.org/licenses/by-nc-nd/2.5/br/`