Construção de Compiladores
Período Especial
Aula 11: Chamadas de Procedimento
Parâmetros passados por Valor

Bruno Müller Junior

Departamento de Informática
UFPR

2020

Sintaxe

- A passagem de parâmetros é alvo das regras 14 e 15.
- Exemplo: `procedure p(x:integer; var y:integer);`
- Onde `x` é passado por valor e `y` por referência.
- Esta aula explica a passagem de parâmetros por valor. A próxima irá explicar a passagem por referência.

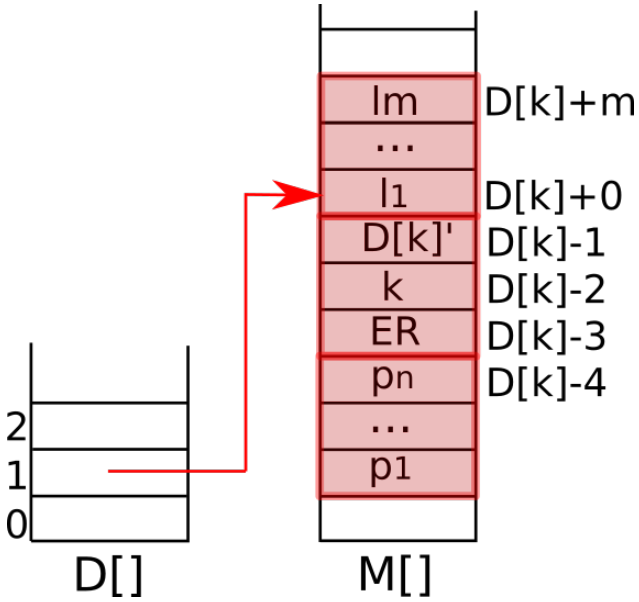
```
12. <declara procedimento> ::= <identificador> [<parâmetros formais>] ;
                               <bloco>
14. <parâmetros formais>   ::= (<seção parâmetros formais>
                             ; <seção parâmetros formais> )

15. <seção parâmetros formais> ::= [VAR] <listaIds>  : <ident>
20. <chamada deprocedimento> ::= <identificador> [(<lista de expressões>)]
```

Nomenclatura

- Para efeito de diferenciação do momento em que são usados, os parâmetros de um procedimento recebem nomes diferentes:
  - **parâmetros formais** variáveis no cabeçalho de um procedimento ou de uma função.
  - **parâmetros reais** variáveis ou constantes que aparecem na chamada dos procedimentos ou das funções.
- Existem duas formas de se passar os parâmetros reais para os parâmetros formais: por valor ou por referência.
  - **por valor** o valor contido no parâmetro real é copiado para o espaço de memória reservado ao parâmetro formal.
  - **por referência** o endereço do parâmetro real é copiado para o parâmetro formal.

Tradução

- O último parâmetro está sempre no deslocamento `k-4`, o penúltimo em `k-5` e assim por diante.
- `procedure p(x:integer; var y:integer);`
  - `y`: `k-4`
  - `x`: `k-5`.
- TS: PF=VS + tipo de passagem (valor ou referência).



```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```

```
program proc2 (input, output);          INPP
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```

```
INPP
AMEM 2
```

Símb.	Cat.	Infos
x	VS	[0,0,int]
y	VS	[0,1,int]

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```

```
        INPP
        AMEM 2
        DSVS R00
R01:ENPR 1
```

```
  p      PROC [1,R01,?{}]
  y      VS   [0,1,int]
  x      VS   [0,0,int]
Símb.  Cat.  Infos
```

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```

```
        INPP
        AMEM 2
        DSVS R00
R01:ENPR 1
```

```
  t      PF    [1,?,?]
  p      PROC  [1,R01,?{}]
  y      VS    [0,1,int]
  x      VS    [0,0,int]
Símb.   Cat.   Infos
```

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```

```
        INPP
        AMEM 2
        DSVS R00
R01:ENPR 1
```

```
  t      PF    [1,?,int]
  p      PROC  [1,R01,?{}]
  y      VS    [0,1,int]
  x      VS    [0,0,int]
Símb.   Cat.   Infos
```

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```

```
        INPP
        AMEM 2
        DSVS R00
R01:ENPR 1
```

Símb.	Cat.	Infos
t	PF	[1,-4,int]
p	PROC	[1,R01,1{int,valor}]
y	VS	[0,1,int]
x	VS	[0,0,int]

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```

```
      INPP
      AMEM 2
      DSVS R00
R01:ENPR 1
      AMEM 1
```

```
  z      VS   [1,0,int]
  t      PF   [1,-4,int]
  p      PROC [1,R01,1{int,valor}]
  y      VS   [0,1,int]
  x      VS   [0,0,int]
Símb.  Cat.  Infos
```

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```

```
      INPP
      AMEM 2
      DSVS R00
R01:ENPR 1
      AMEM
```

```
  z      VS    [1,0,int]
  t      PF    [1,-4,int]
  p      PROC  [1,R01,1{int,valor}]
  y      VS    [0,1,int]
  x      VS    [0,0,int]
Símb.   Cat.   Infos
```

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```

```
     INPP          R02: NADA
     AMEM 2             CRVL 0,1
     DSVS R00           ARMZ 1,0
R01:ENPR 1              CRVL 1,0
     AMEM 1             CRVL 1,-4
     CRVL 1,-4          MULT
     CRCT 1             ARMZ 0,1
     CMMA
     DSVF R03
     CRVL 1,-4
     CRCT 1
     SUBT
     CHPR R01,1
     DSVS R02  <======????
R03:NADA
     CRCT 1
     ARMZ 0,1
```

```
  z      VS    [1,0,int]
  t      PF    [1,-4,int]
  p      PROC  [1,R01,1{int,valor}]
  y      VS    [0,1,int]
  x      VS    [0,0,int]
Símb.   Cat.   Infos
```

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```

```
     INPP
     AMEM 2
     DSVS R00
R01:ENPR 1
     AMEM 1
     CRVL 1,-4
     CRCT 1
     CMMA
     DSVF R03
     CRVL 1,-4
     CRCT 1
     SUBT
     CHPR R01,1
     DSVS R02
R03:NADA
     CRCT 1
     ARMZ 0,1
```

```
R02: NADA
     CRVL 0,1
     ARMZ 1,0
     CRVL 1,0
     CRVL 1,-4
     MULT
     ARMZ 0,1
     DMEM 1
```

Símb.	Cat.	Infos
~~z~~	~~VS~~	~~[1,0,int]~~
t	PF	[1,-4,int]
p	PROC	[1,R01,1{int,valor}]
y	VS	[0,1,int]
x	VS	[0,0,int]

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```

```
     INPP
     AMEM 2
     DSVS R00
R01:ENPR 1
     AMEM 1
     CRVL 1,-4
     CRCT 1
     CMMA
     DSVF R03
     CRVL 1,-4
     CRCT 1
     SUBT
     CHPR R01,1
     DSVS R02
R03:NADA
     CRCT 1
     ARMZ 0,1
```

```
R02: NADA
     CRVL 0,1
     ARMZ 1,0
     CRVL 1,0
     CRVL 1,-4
     MULT
     ARMZ 0,1
     DMEM 1
     RTPR 1,1
```

Símb.	Cat.	Infos
~~t~~	~~PF~~	~~[1,-4,int]~~
p	PROC	[1,R01,1{int,valor}]
y	VS	[0,1,int]
x	VS	[0,0,int]

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```

```
      INPP
      AMEM 2
      DSVS R00
R01:ENPR 1
      AMEM 1
      CRVL 1,-4
      CRCT 1
      CMMA
      DSVF R03
      CRVL 1,-4
      CRCT 1
      SUBT
      CHPR R01,1
      DSVS R02
R03:NADA
      CRCT 1
      ARMZ 0,1
```

```
R02: NADA
      CRVL 0,1
      ARMZ 1,0
      CRVL 1,0
      CRVL 1,-4
      MULT
      ARMZ 0,1
      DMEM 1
      RTPR 1,1
R00:NADA
    LEIT
    ARMZ 0,0
    CRVL 0,0
    CHPR R01,0
    CRVL 0,0
    IMPR
    CRVL 0,1
    IMPR
    DMEM 2
    PARA
```

~~p PROC [1,R01,1{int,valor}]~~
~~y VS [0,1,int]~~
~~x VS [0,0,int]~~

Símb. Cat. Infos

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```

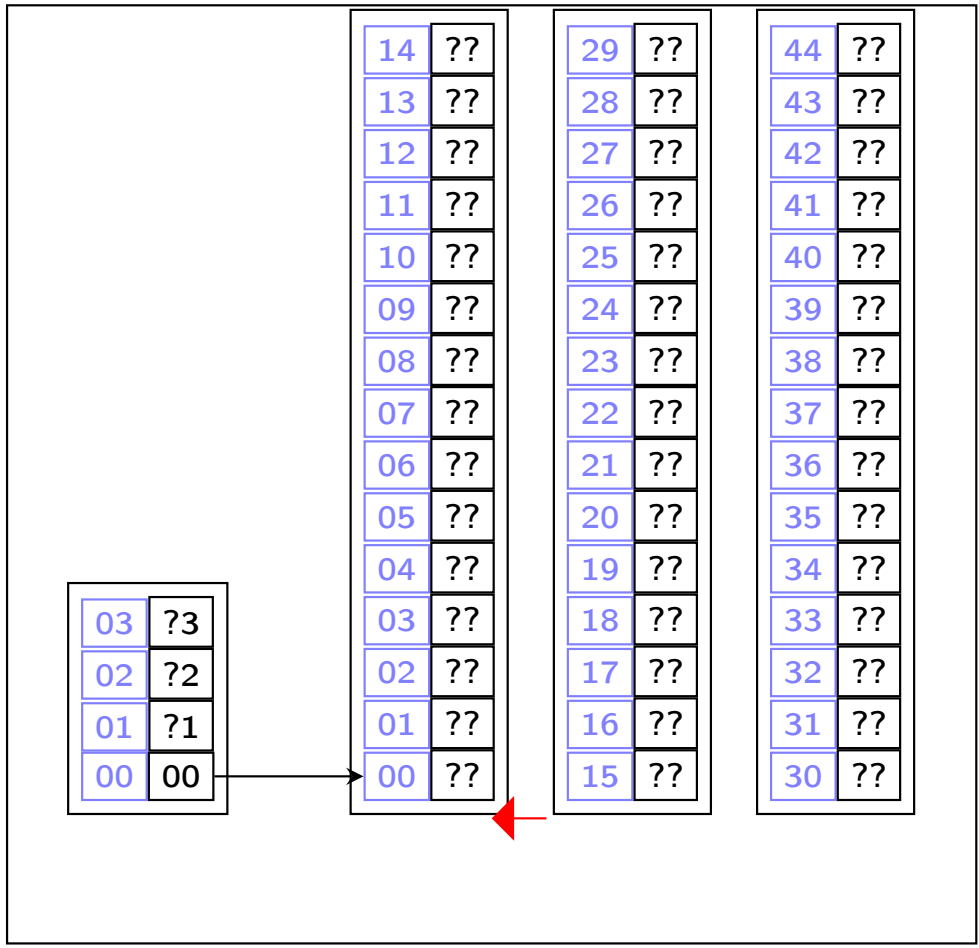
```
     INPP
     AMEM 2
     DSVS R00
R01:ENPR 1
     AMEM 1
     CRVL 1,-4
     CRCT 1
     CMMA
     DSVF R03
     CRVL 1,-4
     CRCT 1
     SUBT
     CHPR R01,1
     DSVS R02
R03:NADA
     CRCT 1
     ARMZ 0,1
```

```
R02: NADA
     CRVL 0,1
     ARMZ 1,0
     CRVL 1,0
     CRVL 1,-4
     MULT
     ARMZ 0,1
     DMEM 1
     RTPR 1,1
R00:NADA
    LEIT
    ARMZ 0,0
    CRVL 0,0
    CHPR R01,0
    CRVL 0,0
    IMPR
    CRVL 0,1
    IMPR
    DMEM 2
    PARA          Símb.   Cat.  Infos
```

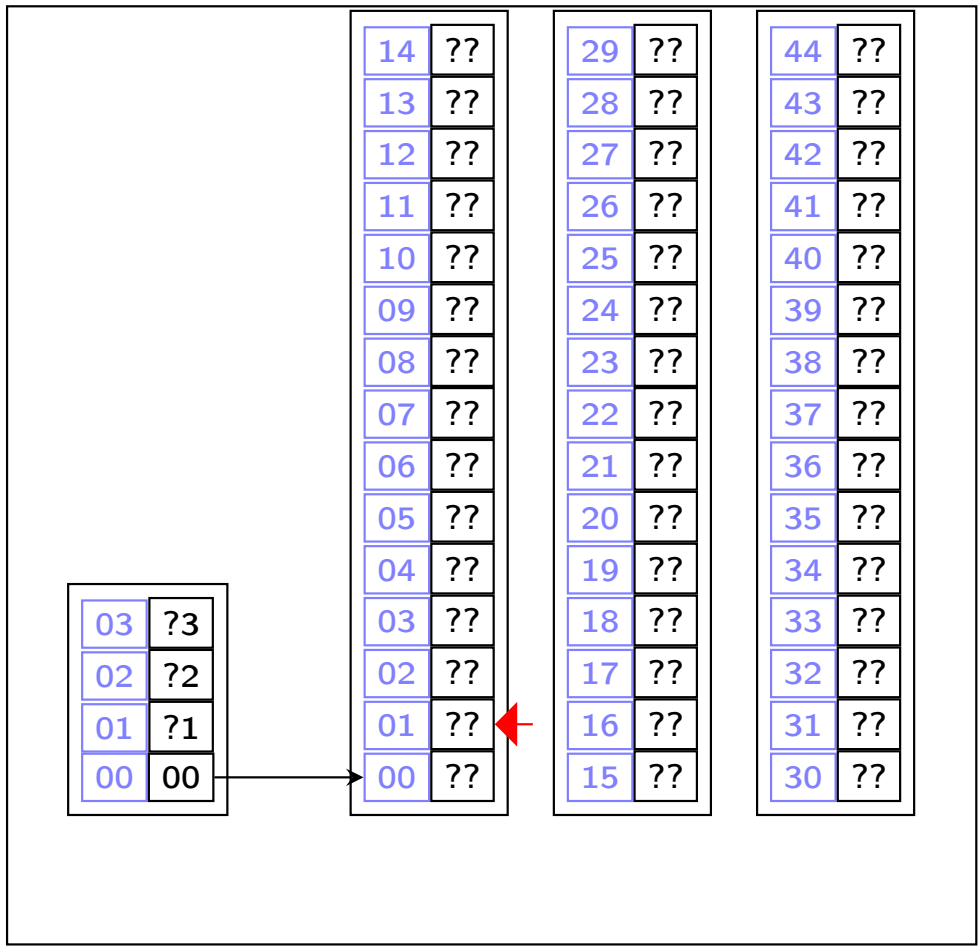
Execução

Principal

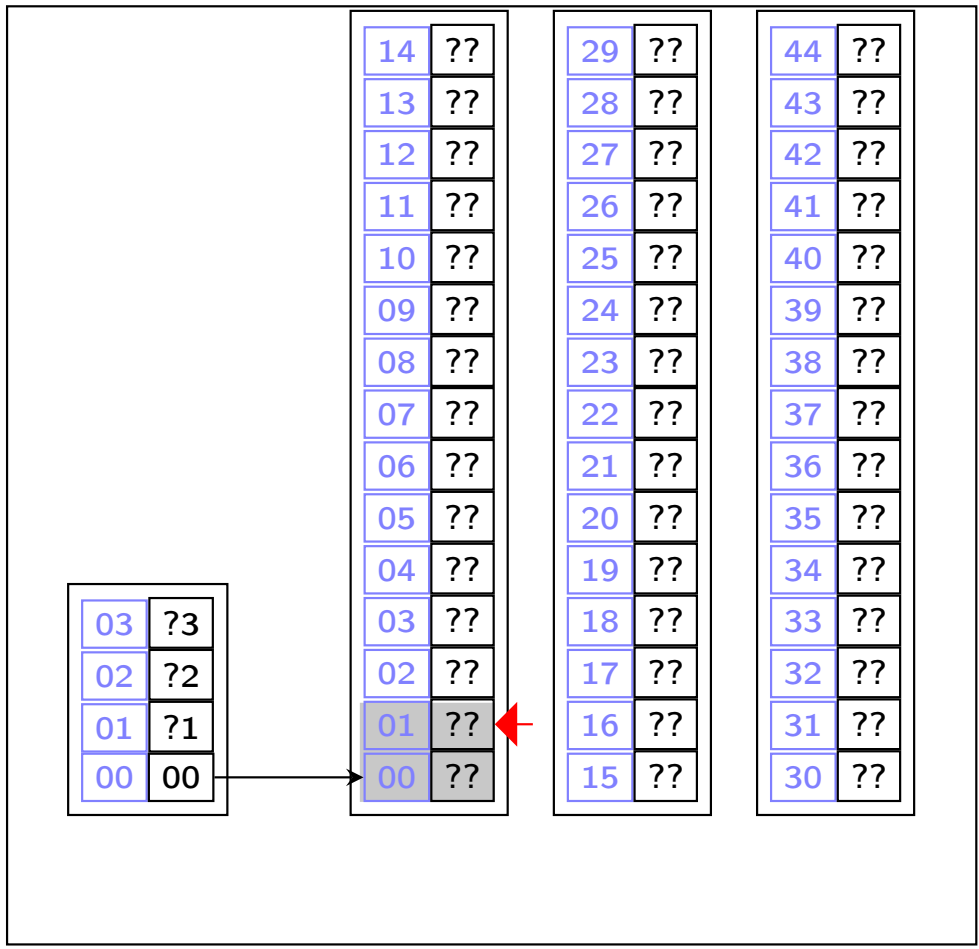
```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```



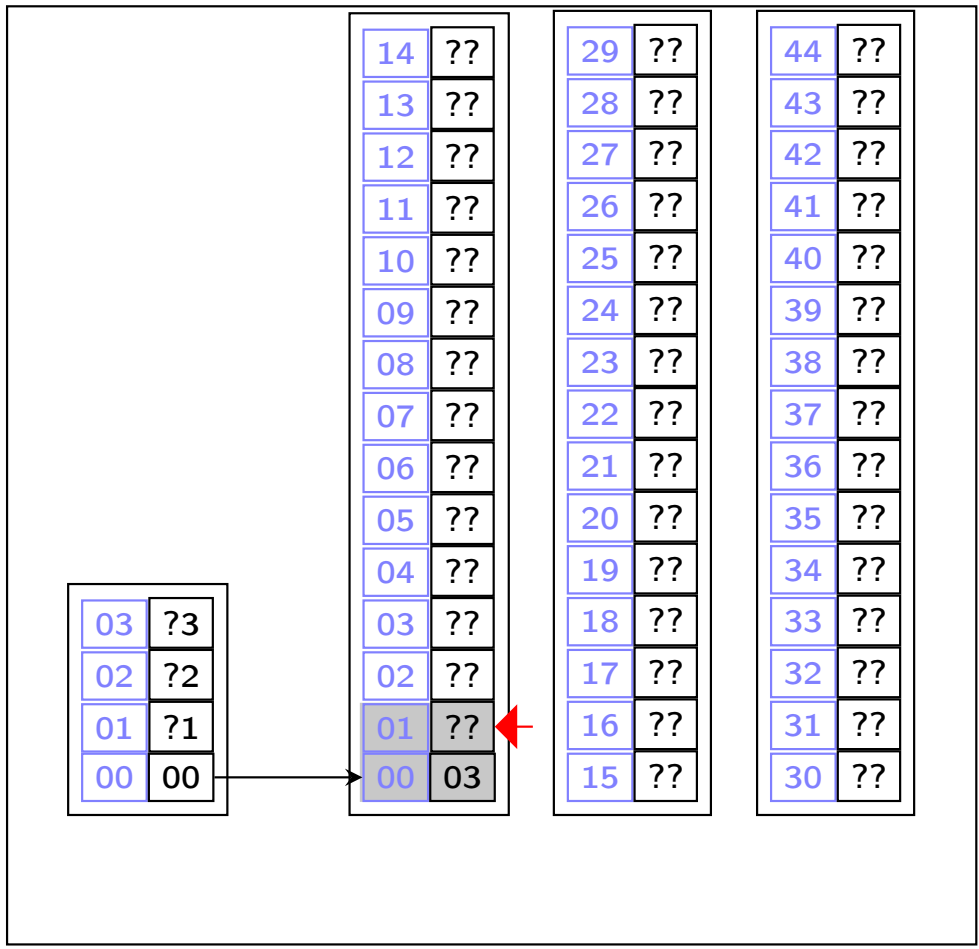
```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```



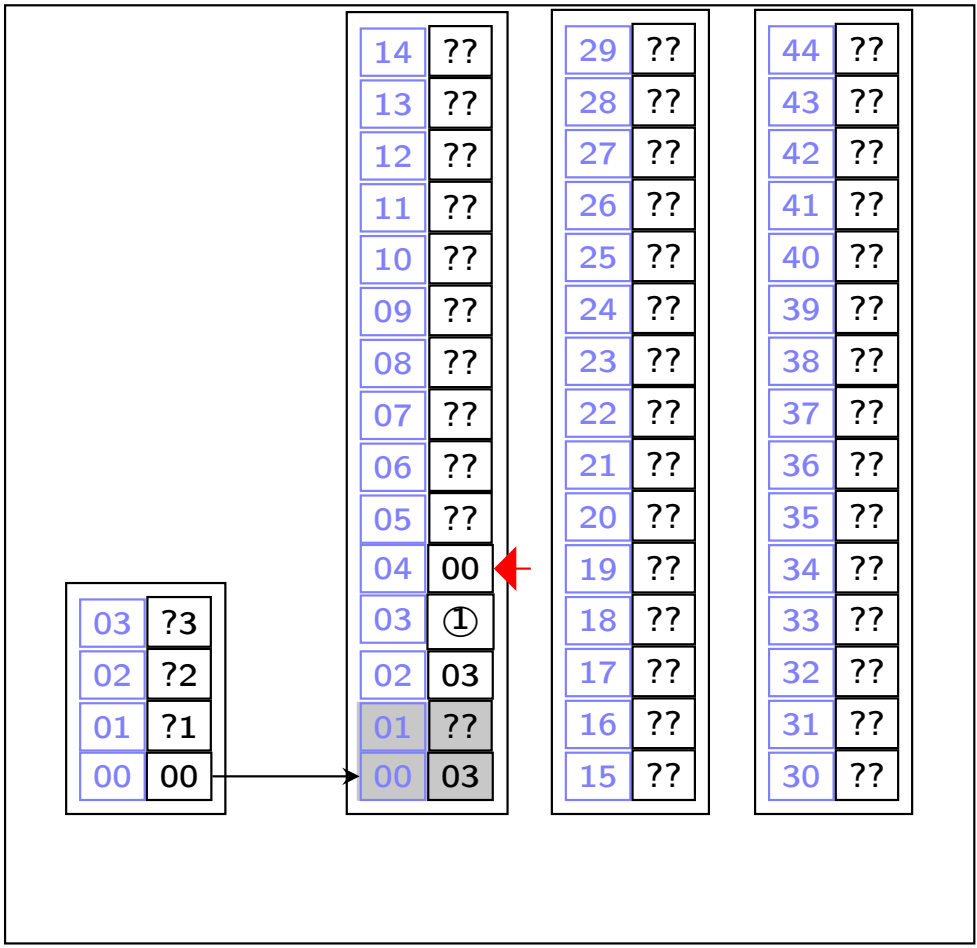
```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```



```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);
   write (x,y)
end.
```

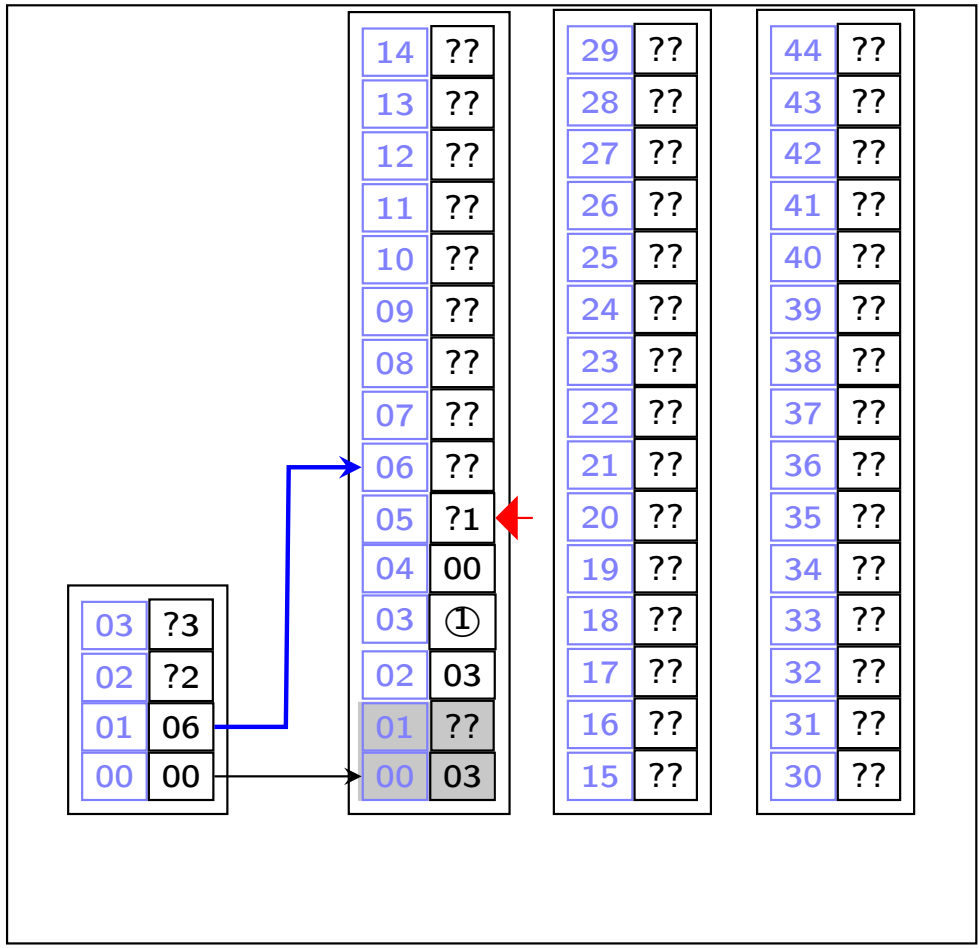


```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
CRVL 0,0
CHPR rot,k { M[s+1]:=i;
             M[s+2]:=k;
             s:=s+2
             i:=rot}
```



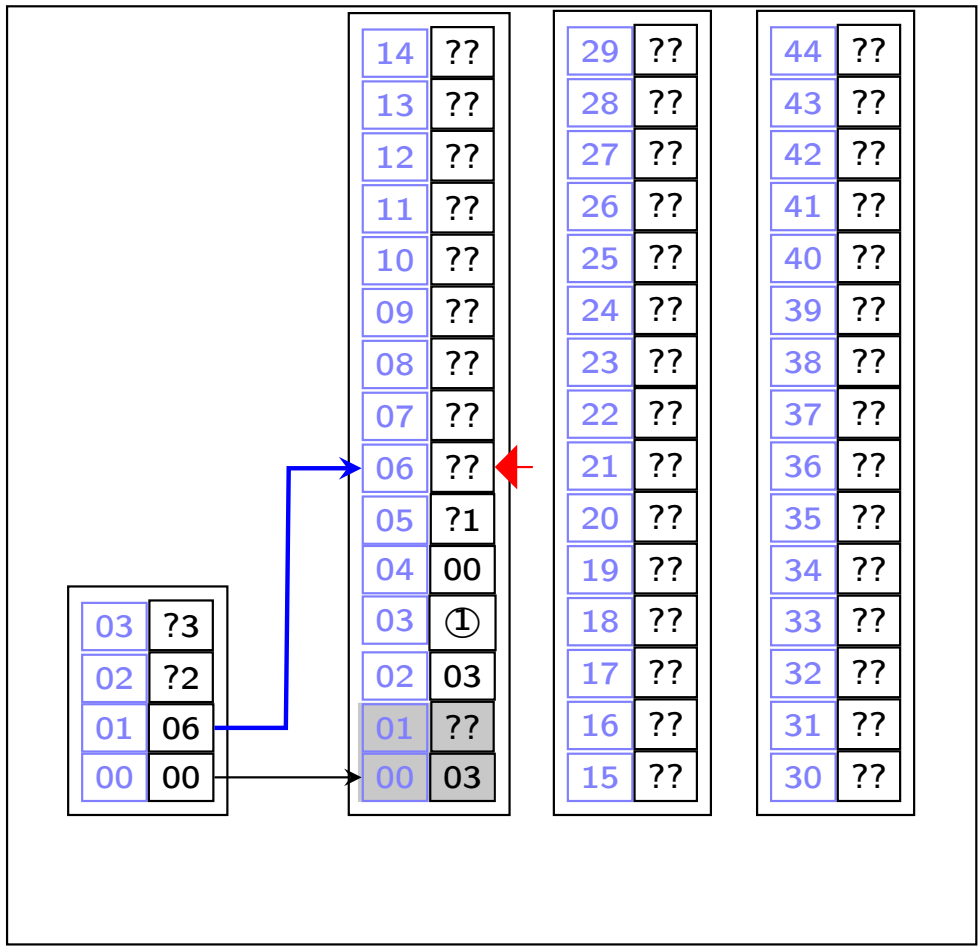
p(3) -> p(2) -> p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.

ENPR k     { s:=s+1;
             M[s]:=D[k]
             D[k]:=s+1 }
```



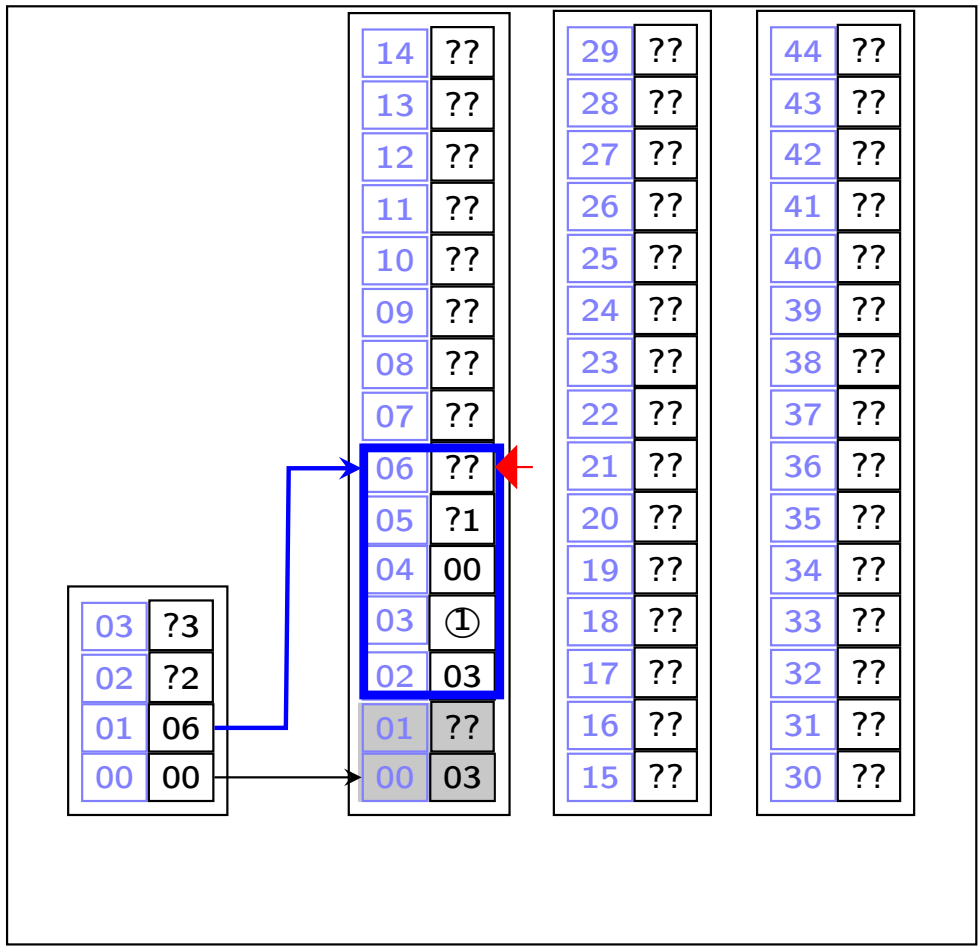
p(3) -> p(2) -> p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
```



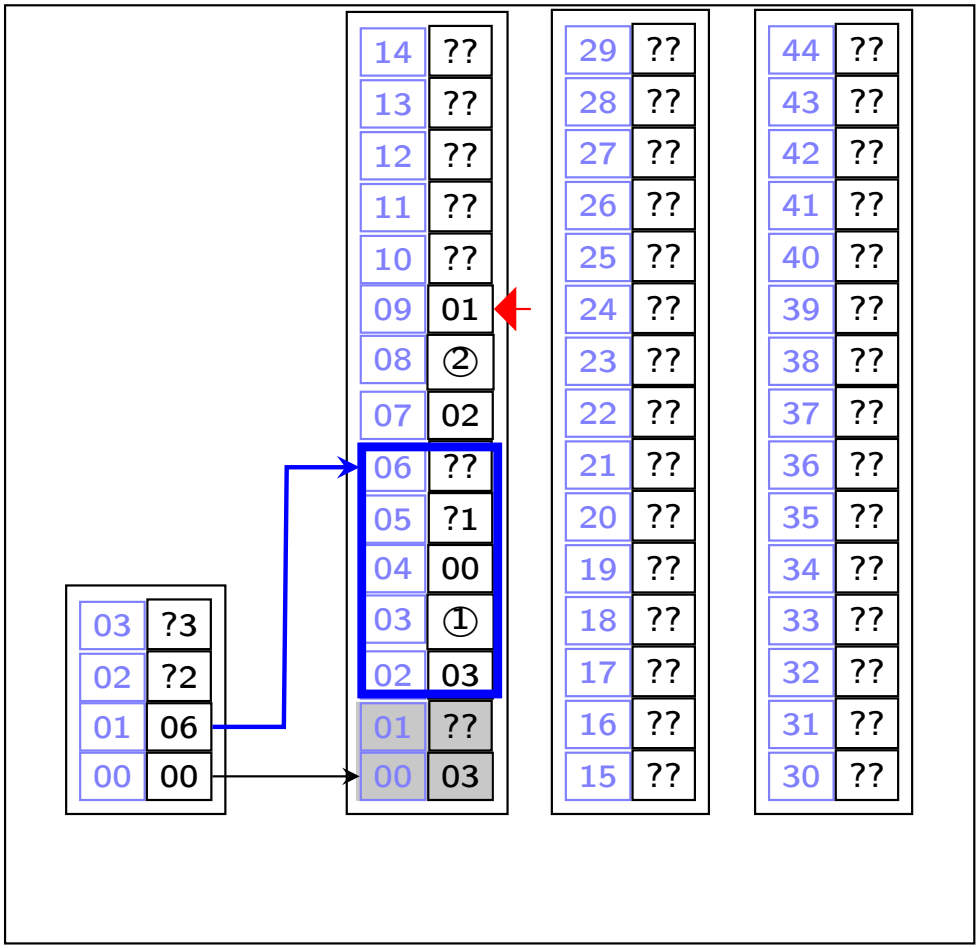
p(3) -> p(2) -> p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
```



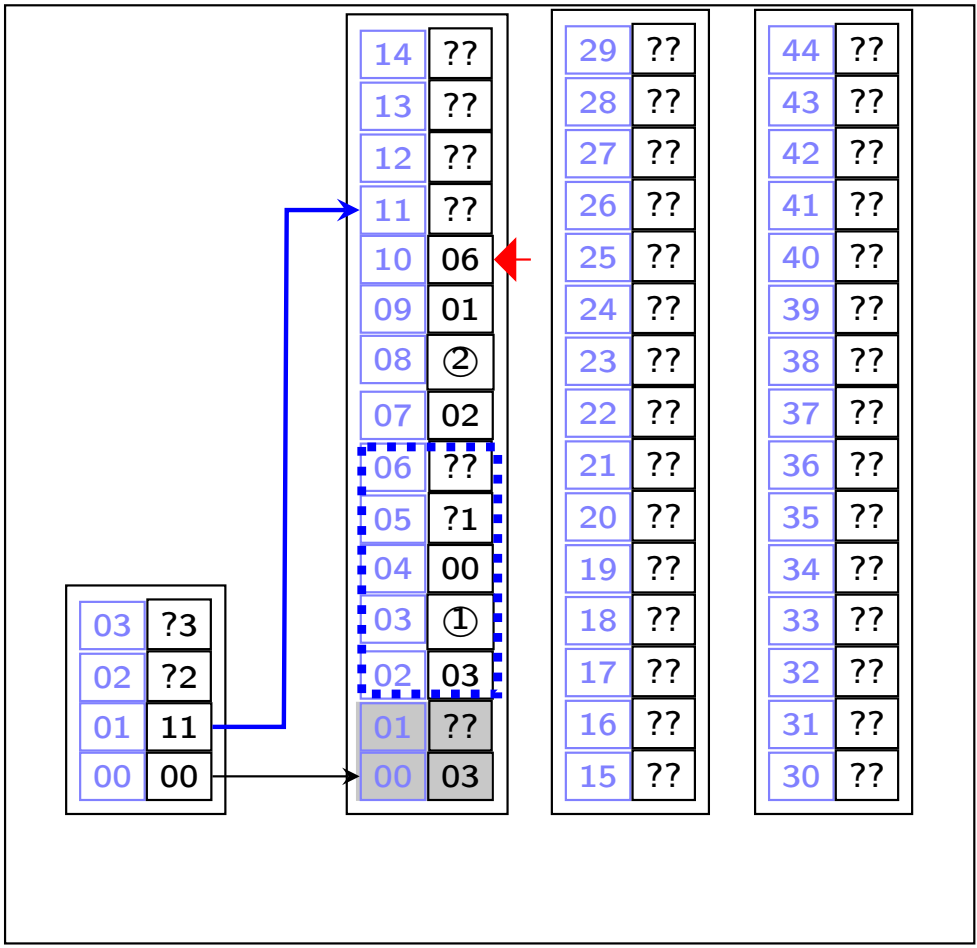
p(3) -> p(2) -> p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
CRVL 0,0 / CRCT 1 / SUBT
CHPR rot,k { M[s+1]:=i;
             M[s+2]:=k;
             s:=s+2
             i:=rot}
```



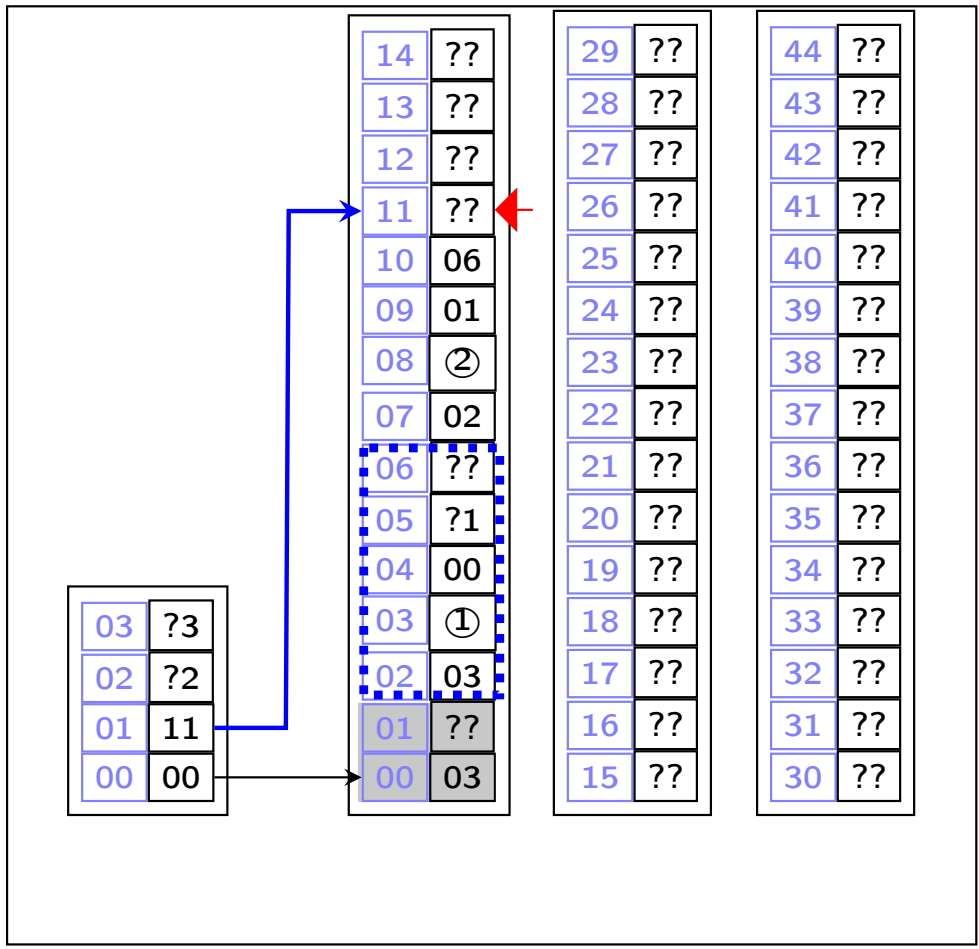
p(3) -> p(2) -> p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.

ENPR k      { s:=s+1;
              M[s]:=D[k]
              D[k]:=s+1 }
```



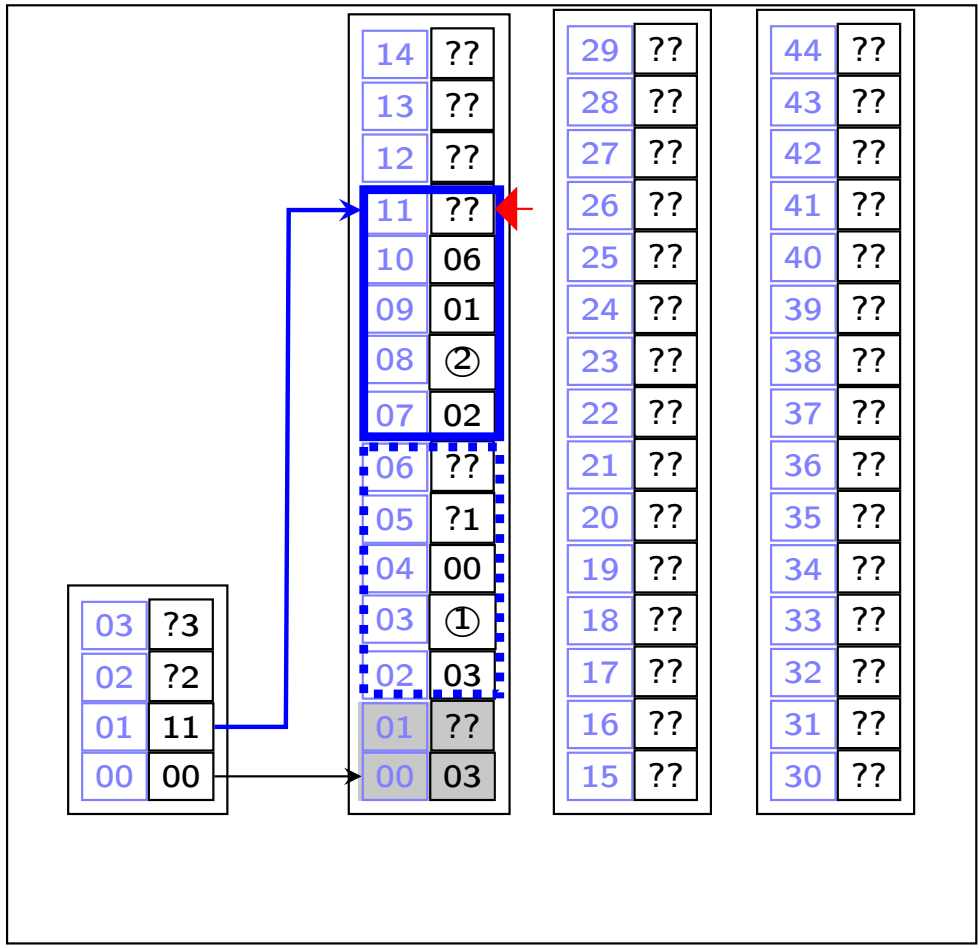
p(3) -> p(2) -> p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
```



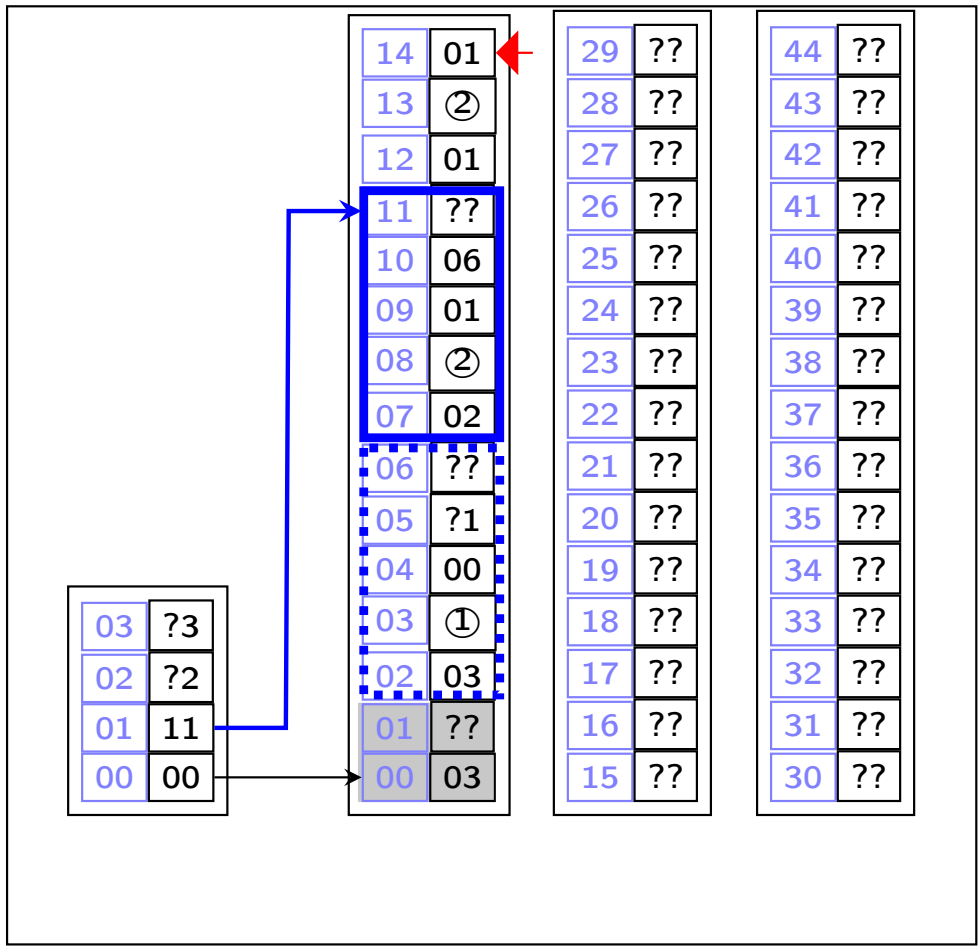
p(3) -> p(2) -> p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
```



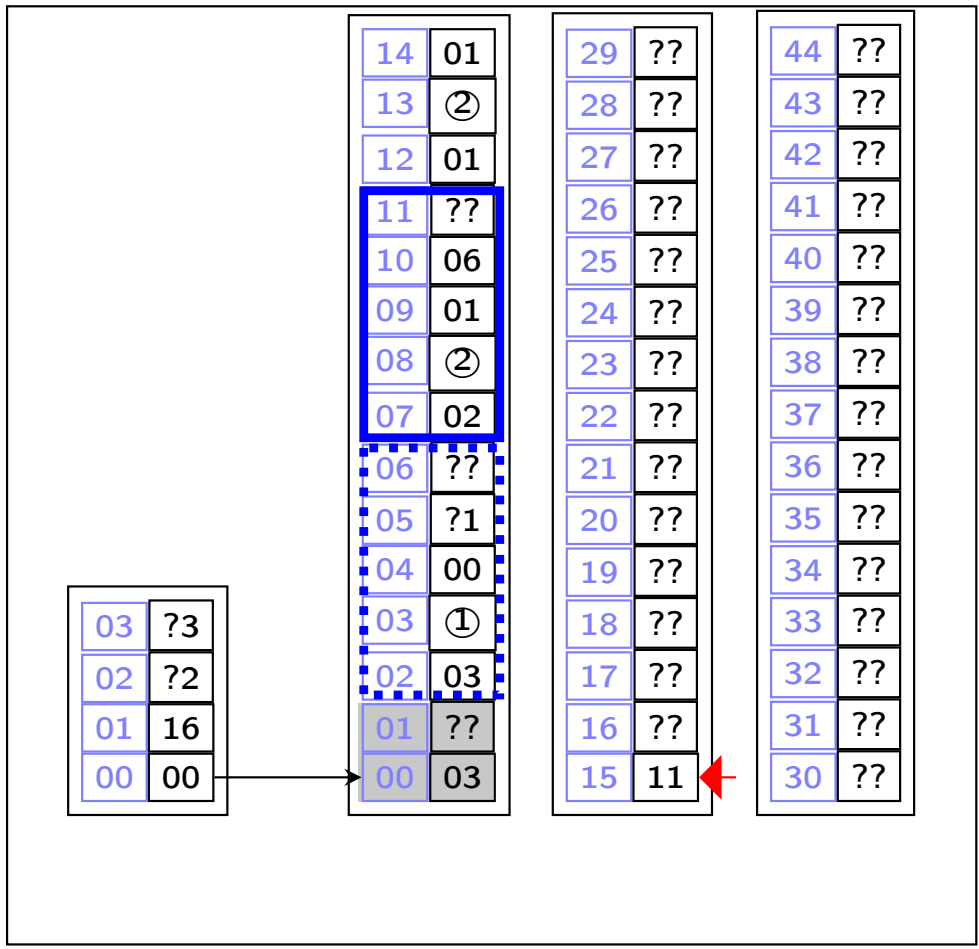
p(3) -> p(2) -> p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
CRVL 0,0 / CRCT 1 / SUBT
CHPR rot,k { M[s+1]:=i;
             M[s+2]:=k;
             s:=s+2
             i:=rot}
```



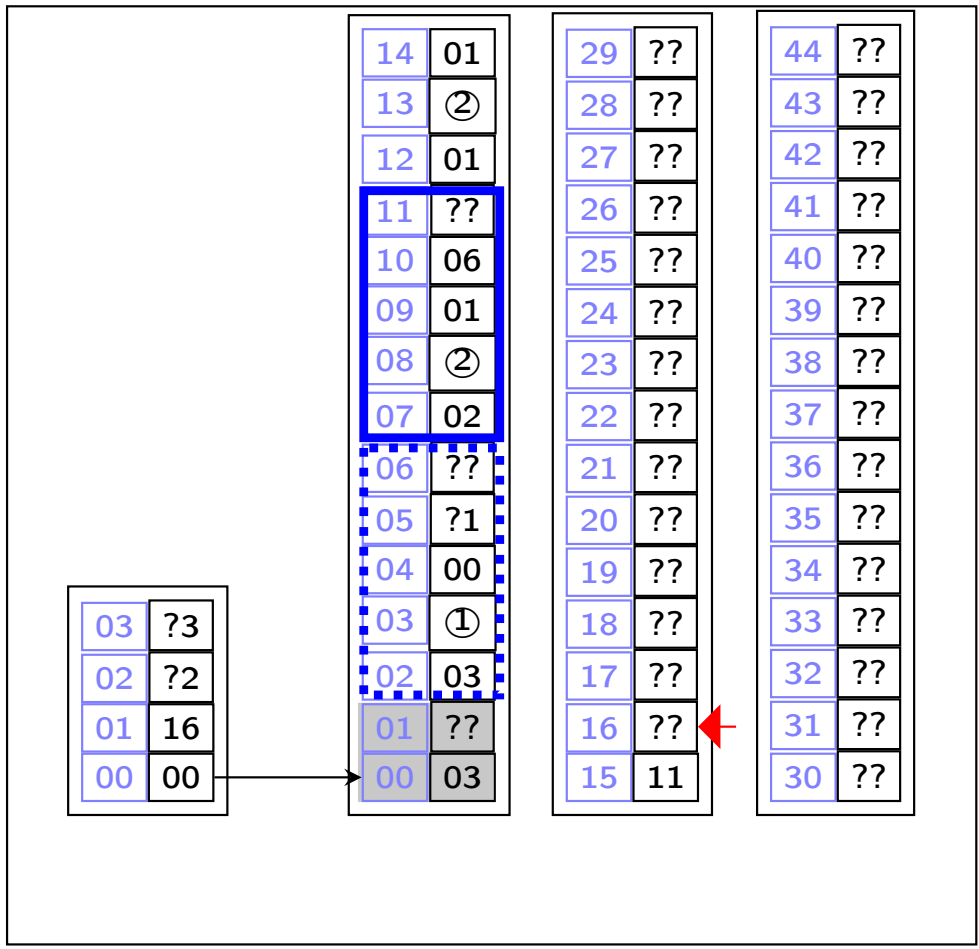
p(3) -> p(2) -> p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.

ENPR k      { s:=s+1;
              M[s]:=D[k]
              D[k]:=s+1 }
```



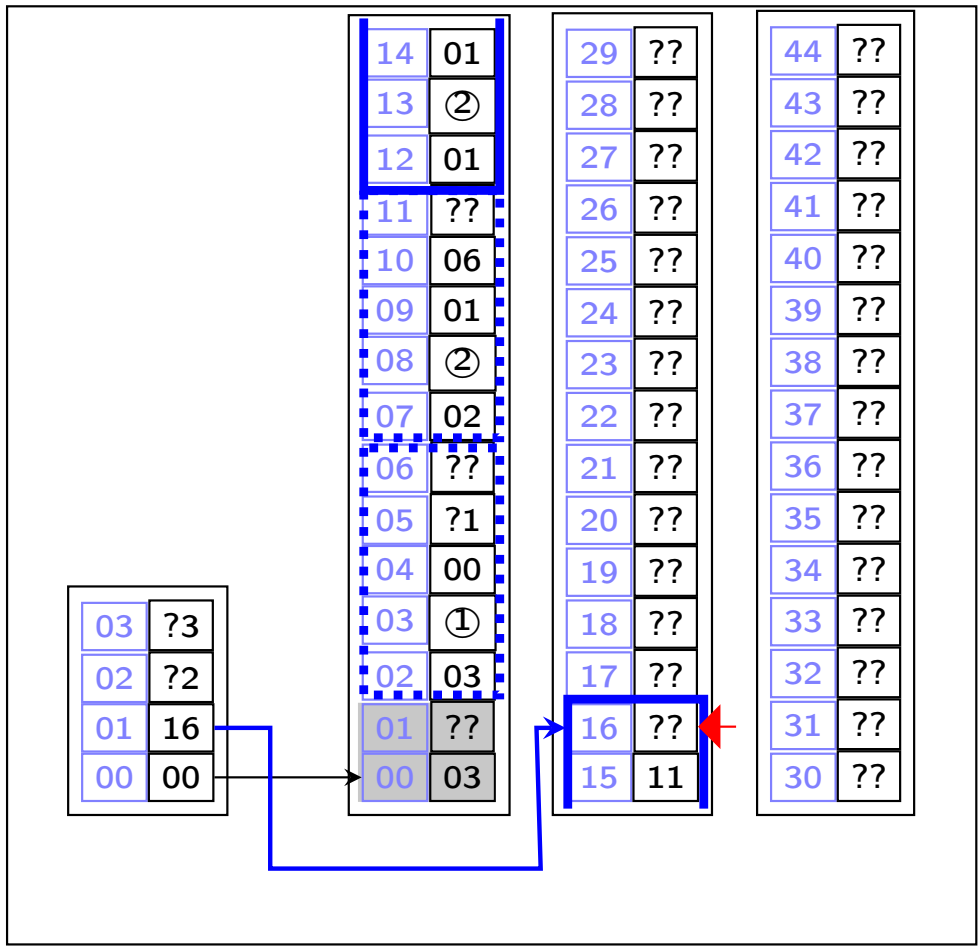
p(3) -> p(2) -> p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
```



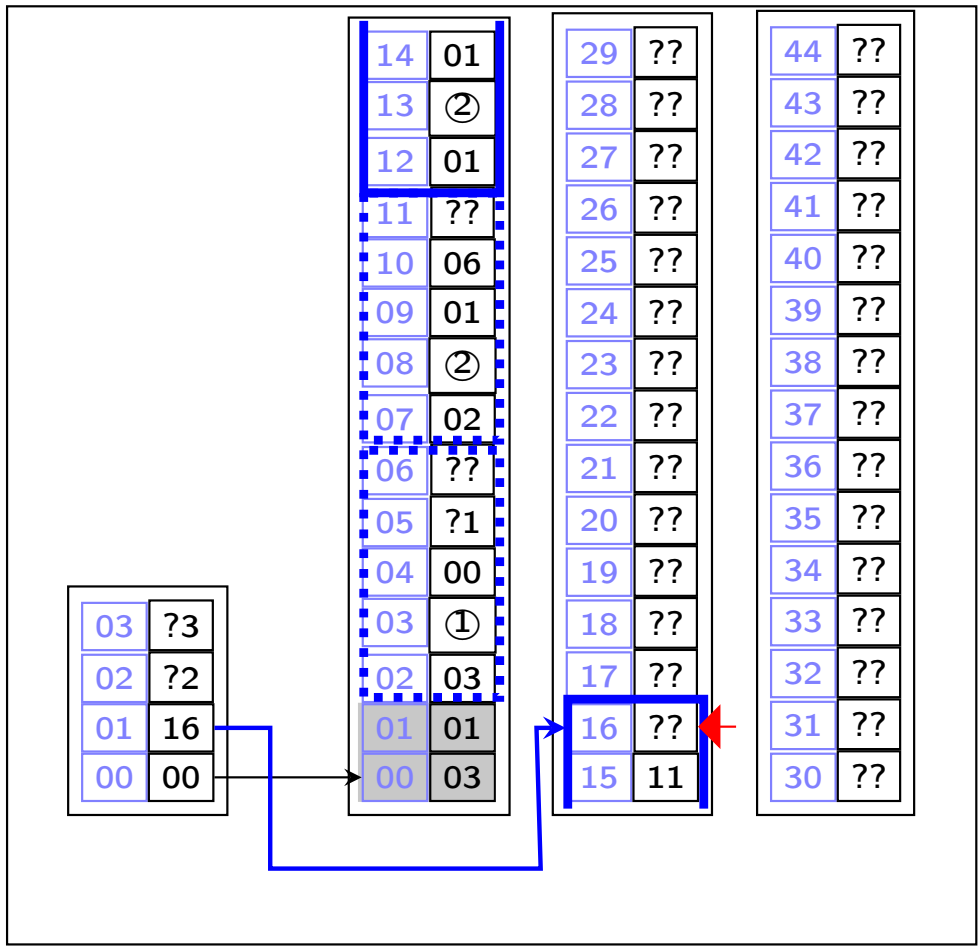
p(3) -> p(2) -> p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
```



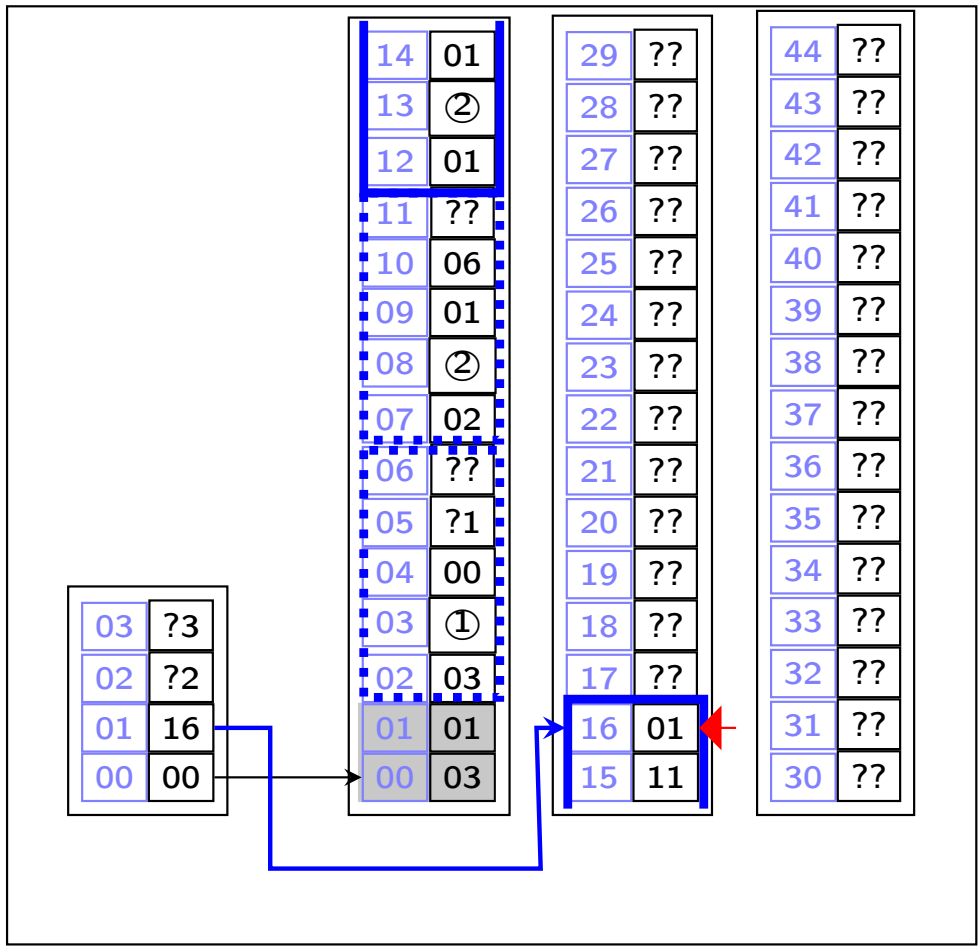
p(3) -> p(2) -> p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
```



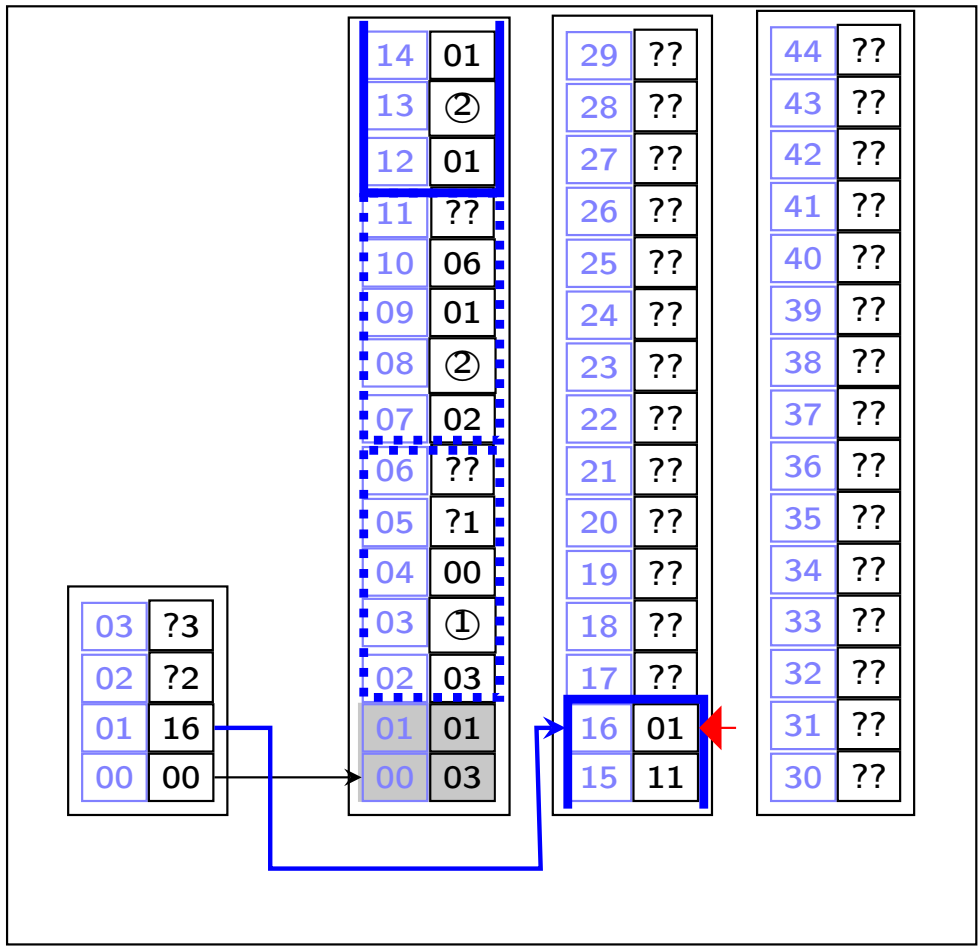
p(3) -> p(2) -> p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
```



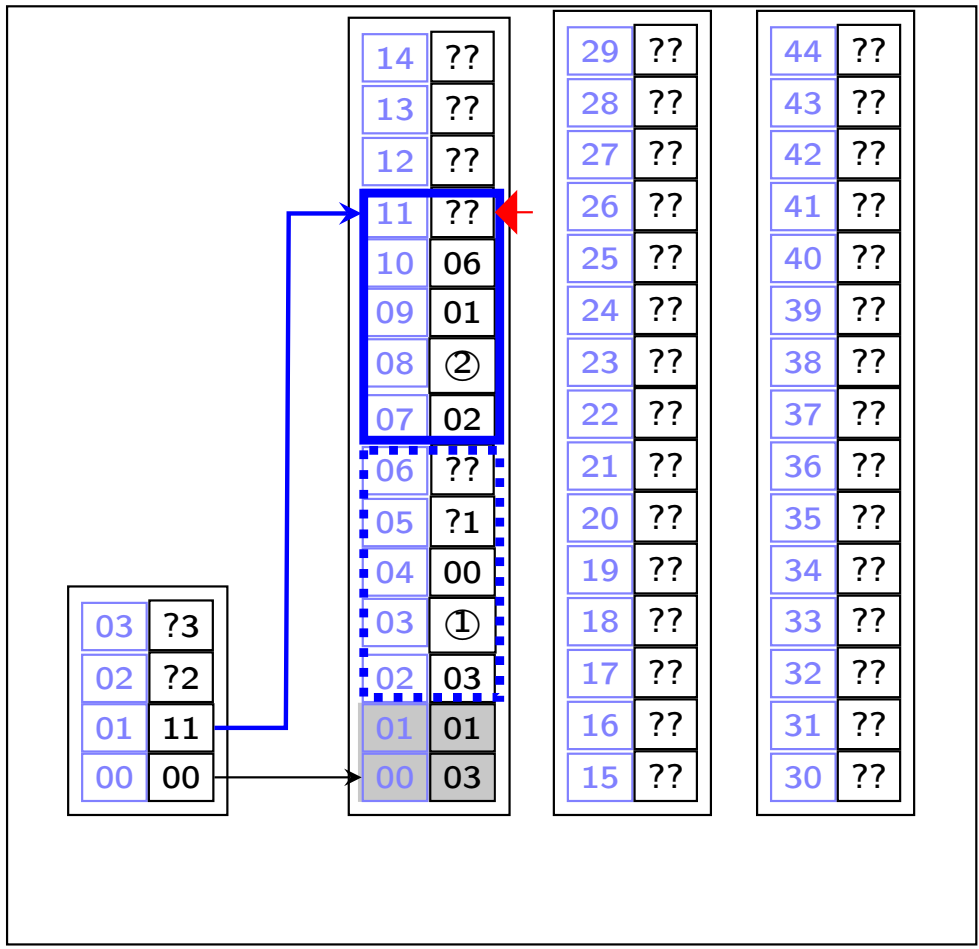
p(3) -> p(2) -> p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
```



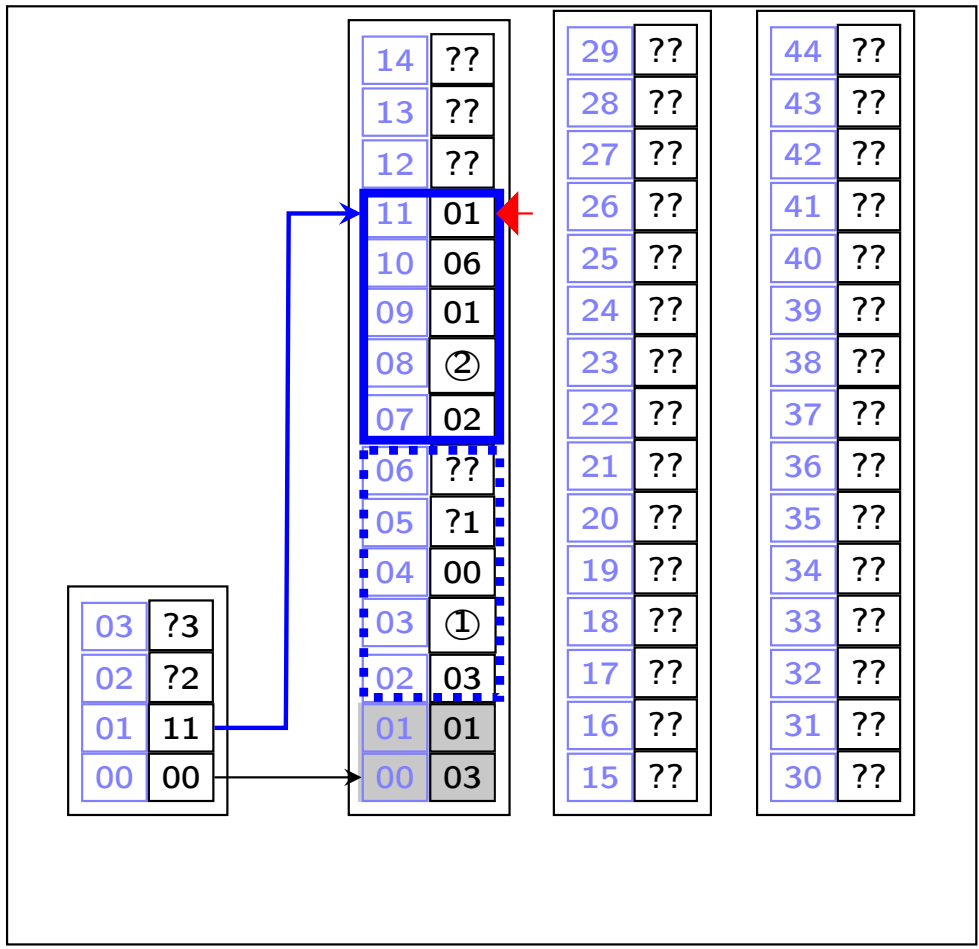
princ <- p(3) <- p(2) <- p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.

DMEM 1
RTPR k,n { D[K]:=M[s];
           i:=M[s-2];
           s:=s-(n+3)}
```



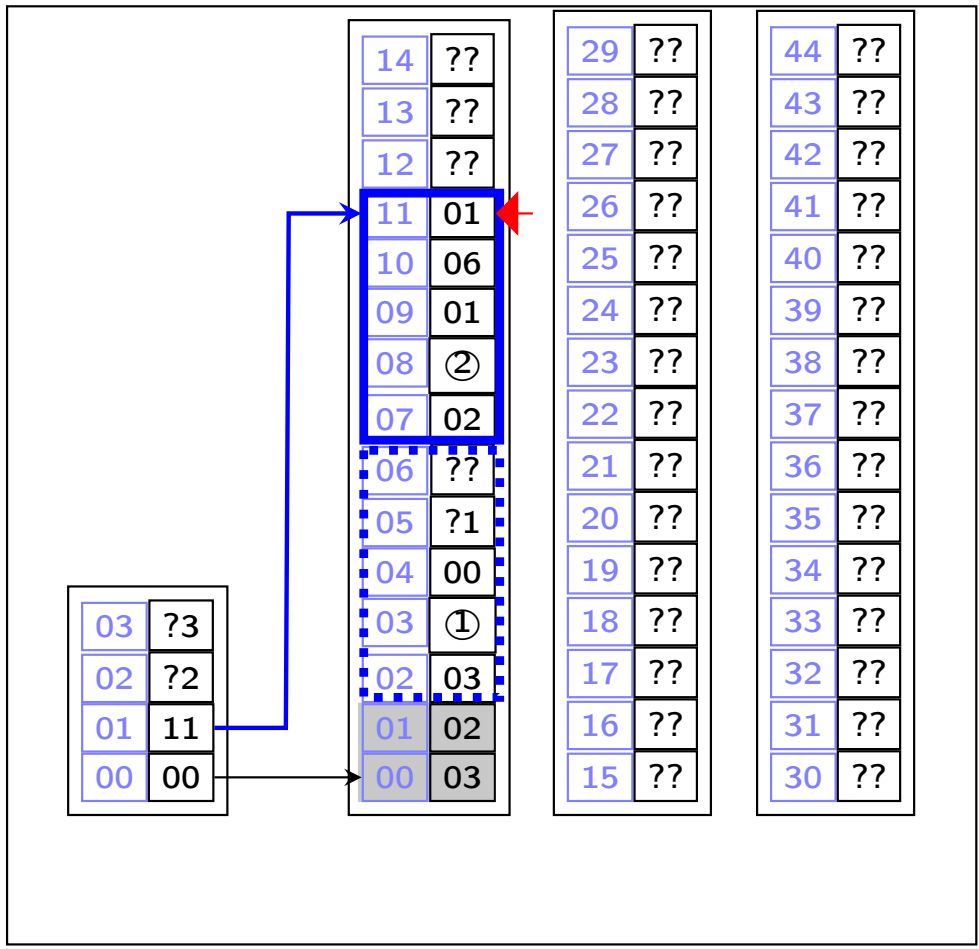
princ <- p(3) <- p(2) <- p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
```



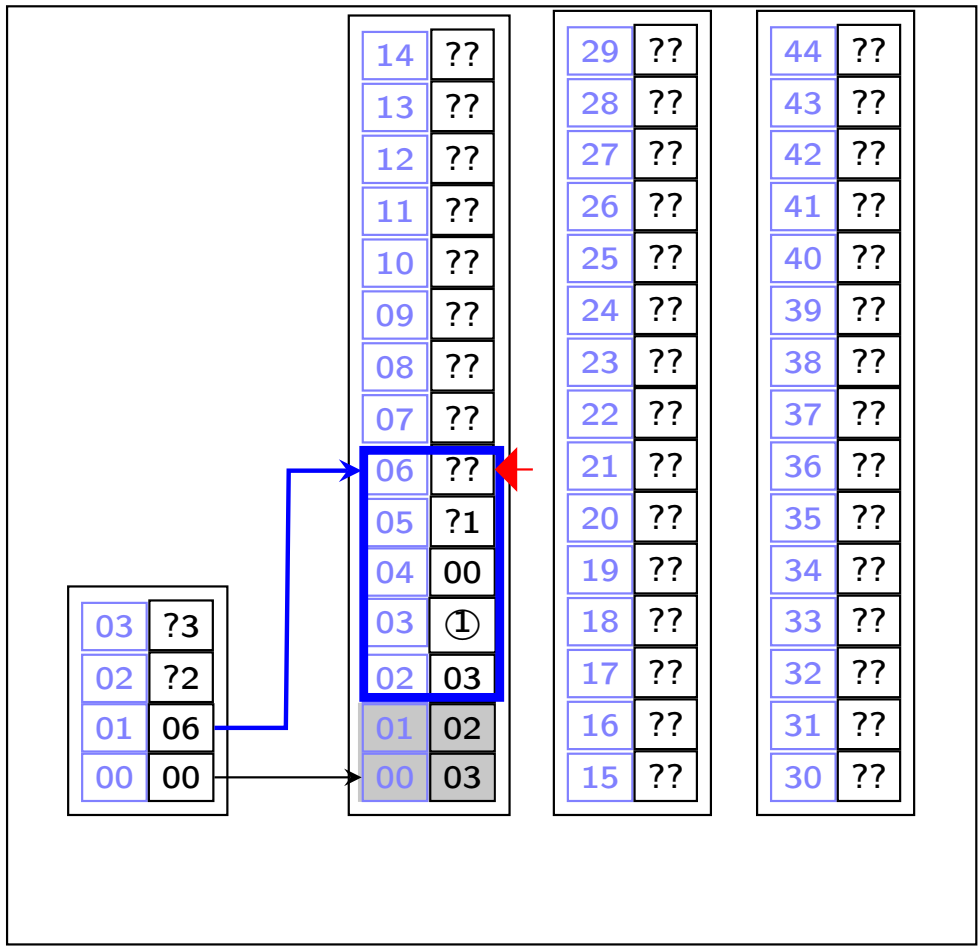
princ <- p(3) <- p(2) <- p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
```



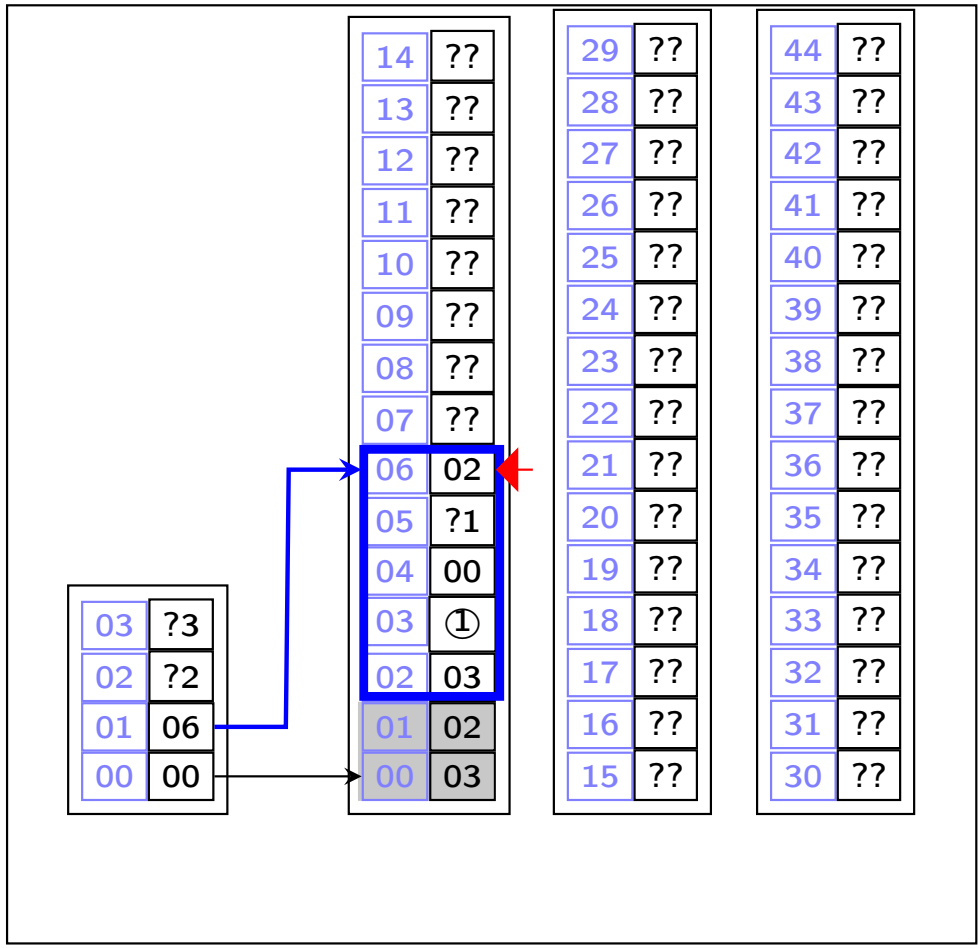
princ <- p(3) <- p(2) <- p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.

DMEM 1
RTPR k,n { D[K]:=M[s];
           i:=M[s-2];
           s:=s-(n+3)}
```



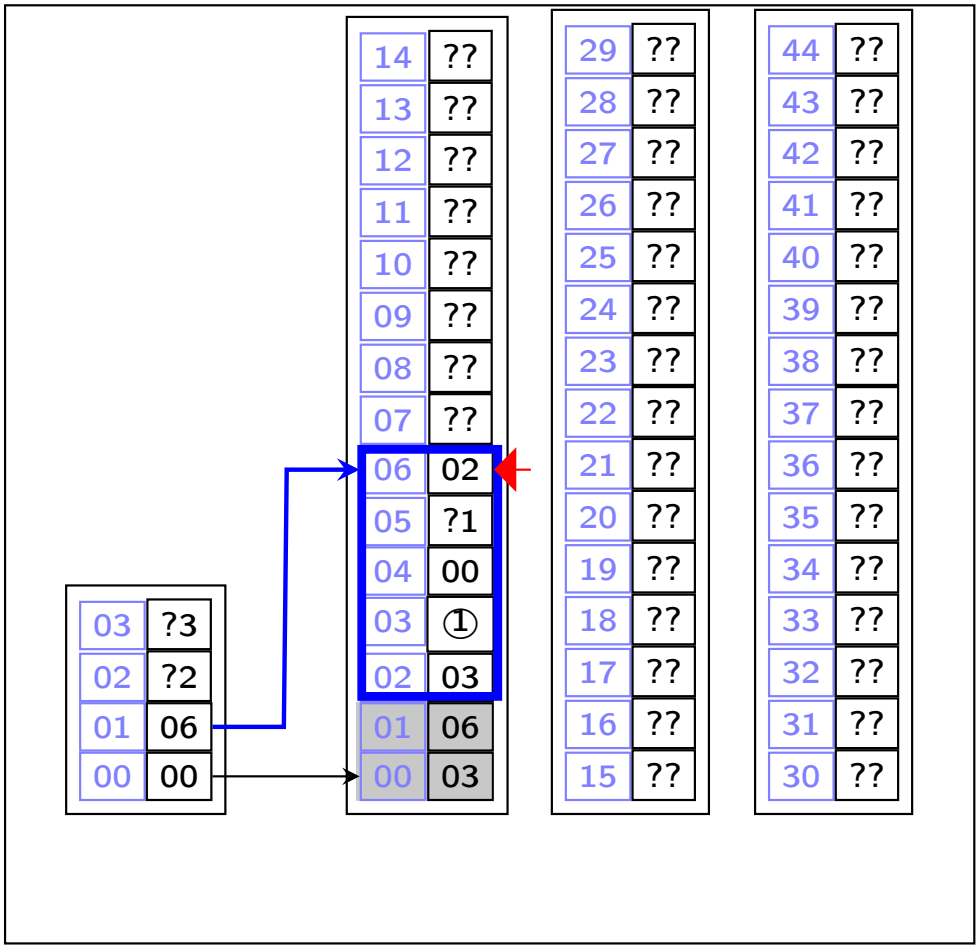
princ <- p(3) <- p(2) <- p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
```



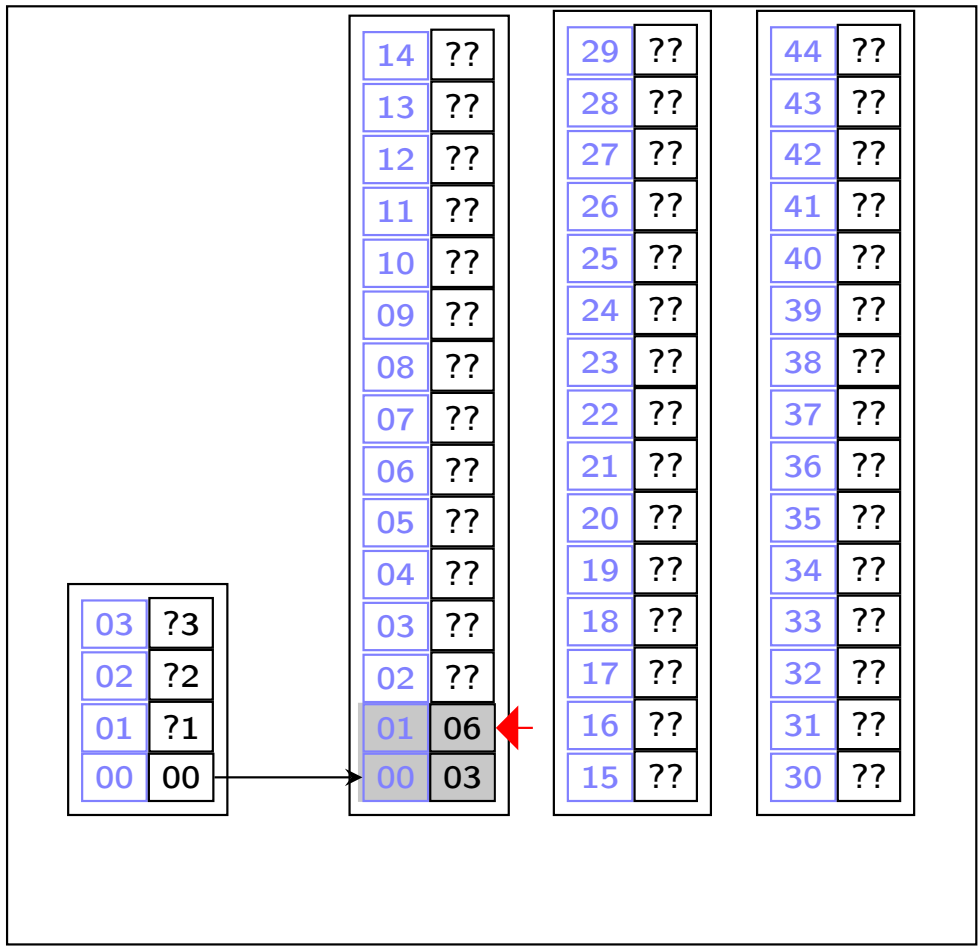
princ <- p(3) <- p(2) <- p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
```



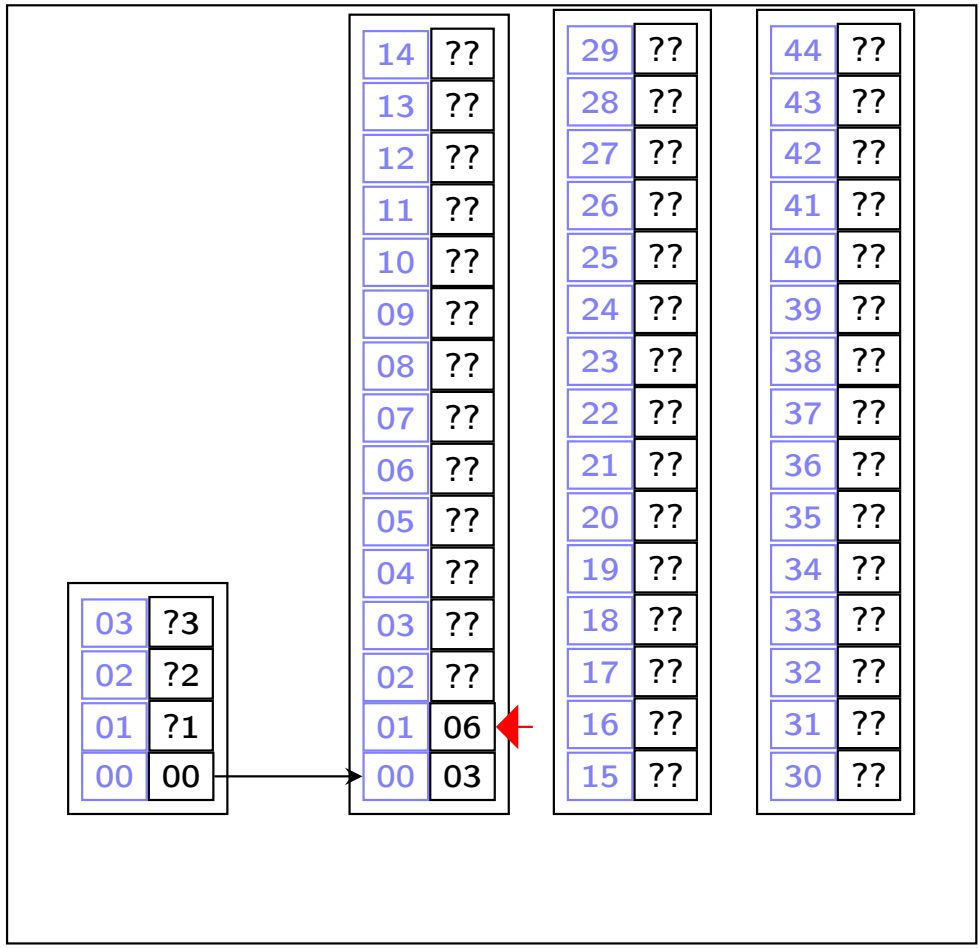
princ <- p(3) <- p(2) <- p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.

DMEM 1
RTPR k,n { D[K]:=M[s];
           i:=M[s-2];
           s:=s-(n+3)}
```



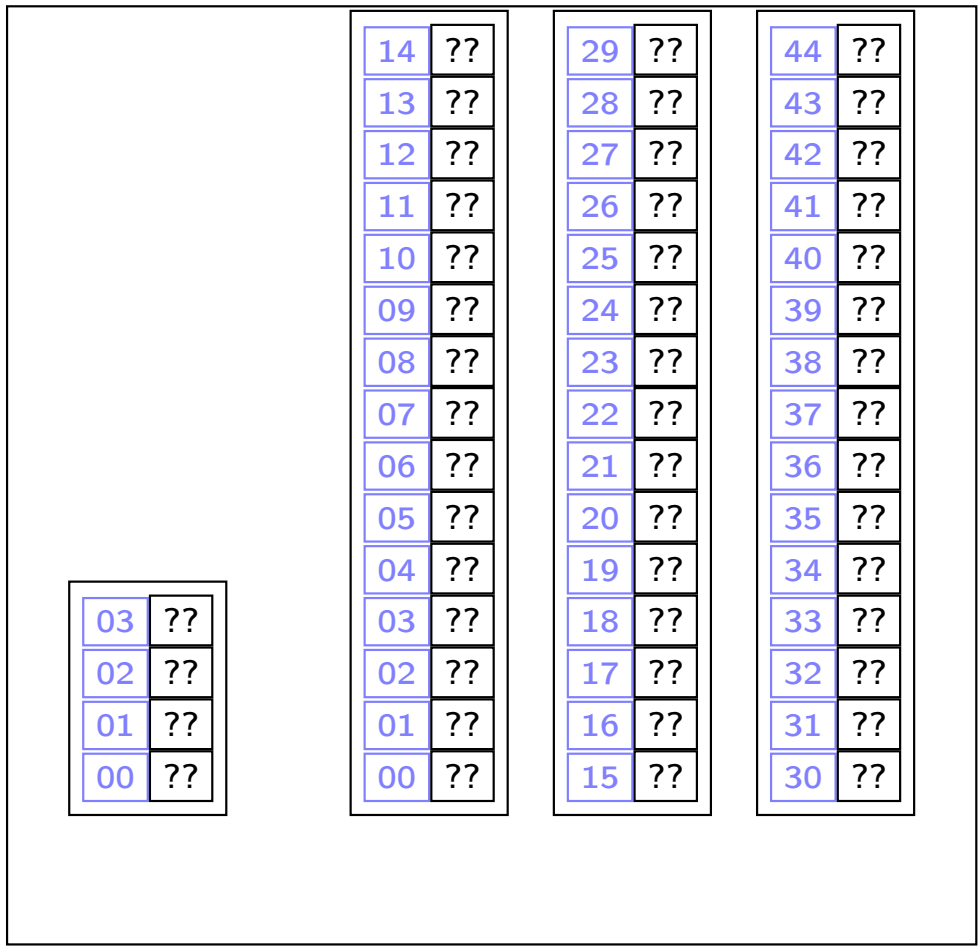
princ <- p(3) <- p(2) <- p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
```



princ <- p(3) <- p(2) <- p(1)

```
program proc2 (input, output);
var x, y: integer;
   procedure p(t:integer);
   var z:integer;
      begin
        if (t>1)
           then p(t-1);②
           else y:=1;
        z:= y;
        y:=z*t;
     end
begin
   read(x);
   p(x);①
   write (x,y)
end.
```



- Acrescente a chamada de procedimentos com parâmetros passados por valor ao compilador.
- No exemplo apresentado nesta aula, só há um parâmetro.
- Quando houverem mais parâmetros, determinar o endereço léxico é um pouco mais complicado, pois o último parâmetro é quem será associado ao endereço léxico (k-4) Por exemplo:
  - procedure p(a,b:integer), a$\Rightarrow (k,-5)$, b$\Rightarrow (k,-4)$
  - procedure q(a,b,c:integer), a$\Rightarrow (k,-6)$, b$\Rightarrow (k,-5)$, c$\Rightarrow (k,-4)$

- 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/`