

# Adressen und Pointer

# Speicheraufbau:

Speicherbereich  
(beschreibbar)      Adressbereich  
(nur lesbar)

?	←	0000
?	←	0001
?	←	0002
?	←	0003
?	←	0004
?	←	0005
?	←	0006
?	←	0007
?	←	0008
?	←	0009
?	←	0010
?	←	0011

# Speicheraufbau:

Speicherbereich (beschreibbar)      Adressbereich (nur lesbar)

?	←	0000
?	←	0001
?	←	0002
?	←	0003
?	←	0004
?	←	0005
?	←	0006
?	←	0007
?	←	0008
?	←	0009
?	←	0010
?	←	0011

# Quelltext:

Namen (nur Programmierer)

A
Ergebnis
Tmp
var1
Var2

← Compiler

# Programmablauf:

Speicherbereich (beschreibbar)	Adressbereich (nur lesbar)	Namen	
?	← 0000		void main () {
?	← 0001		
?	← 0002		
?	← 0003		
?	← 0004		
?	← 0005		
?	← 0006		
?	← 0007		
?	← 0008		
?	← 0009		
?	← 0010		
?	← 0011		}

# Programmablauf:

Speicherbereich (beschreibbar)	Adressbereich (nur lesbar)	Namen
?	0000	i
?	0001	
?	0002	
?	0003	
?	0004	
?	0005	
?	0006	
?	0007	
?	0008	
?	0009	
?	0010	
?	0011	

```
void main () {  
    int i;  
  
}
```

# Programmablauf:

Speicherbereich (beschreibbar)	Adressbereich (nur lesbar)	Namen
5	0000	i
?	0001	
?	0002	
?	0003	
?	0004	
?	0005	
?	0006	
?	0007	
?	0008	
?	0009	
?	0010	
?	0011	

```
void main () {  
    int i;  
    i = 5;  
}
```

# Programmablauf:

Speicherbereich (beschreibbar)	Adressbereich (nur lesbar)	Namen
5	0000	i
65	0001	j
?	0002	
?	0003	
?	0004	
?	0005	
?	0006	
?	0007	
?	0008	
?	0009	
?	0010	
?	0011	

ASCII 'A' =  
65 d

```
void main() {  
    int i;  
    i = 5;  
    int j = 'A';  
}
```

# Programmablauf:

Speicherbereich (beschreibbar)	Adressbereich (nur lesbar)	Namen
5	0000	i
65	0001	j
?	0002	x
?	0003	
?	0004	
?	0005	
?	0006	
?	0007	
?	0008	
?	0009	
?	0010	
?	0011	

```
void main () {  
    int i;  
    i = 5;  
    int j = 'A';  
  
    int *x;  
  
}
```



# Programmablauf:

Speicherbereich (beschreibbar)	Adressbereich (nur lesbar)	Namen
5	0000	i
65	0001	j
0001	0002	x
?	0003	
?	0004	
?	0005	
?	0006	
?	0007	
?	0008	
?	0009	
?	0010	
?	0011	

```
void main () {  
    int i;  
    i = 5;  
    int j = 'A';  
  
    int *x;  
    x = &j;  
  
}
```

# Programmablauf:

Speicherbereich (beschreibbar)	Adressbereich (nur lesbar)	Namen
5	0000	i
65	0001	j
0001	0002	x
?	0003	
?	0004	
?	0005	
?	0006	
?	0007	
?	0008	
?	0009	
?	0010	
?	0011	

```
void main () {  
    int i;  
    i = 5;  
    int j = 'A';  
  
    int *x;  
    x = &j;  
  
    printf(„%d“, j);  
  
}
```

# Programmablauf:

Speicherbereich (beschreibbar)	Adressbereich (nur lesbar)	Namen
5	0000	i
65	0001	j
0001	0002	x
?	0003	
?	0004	
?	0005	
?	0006	
?	0007	
?	0008	
?	0009	
?	0010	
?	0011	

```
void main () {  
    int i;  
    i = 5;  
    int j = 'A';  
  
    int *x;  
    x = &j;  
  
    printf(„%d“, j);  
        $ 65  
  
}
```

# Programmablauf:

Speicherbereich (beschreibbar)	Adressbereich (nur lesbar)	Namen
5	0000	i
65	0001	j
0001	0002	x
?	0003	
?	0004	
?	0005	
?	0006	
?	0007	
?	0008	
?	0009	
?	0010	
?	0011	

```
void main () {  
    int i;  
    i = 5;  
    int j = 'A';  
  
    int *x;  
    x = &j;  
  
    printf(„%d“, j);  
        $ 65  
    printf(„%d“, x);  
  
}
```

# Programmablauf:

Speicherbereich (beschreibbar)	Adressbereich (nur lesbar)	Namen
5	0000	i
65	0001	j
0001	0002	x
?	0003	
?	0004	
?	0005	
?	0006	
?	0007	
?	0008	
?	0009	
?	0010	
?	0011	

```
void main () {  
    int i;  
    i = 5;  
    int j = 'A';  
  
    int *x;  
    x = &j;  
  
    printf(„%d“, j);  
        $ 65  
    printf(„%d“, x);  
        $ 0001  
  
}
```

# Programmablauf:

Speicherbereich (beschreibbar)	Adressbereich (nur lesbar)	Namen
5	0000	i
65	0001	j
0001	0002	x
?	0003	
?	0004	
?	0005	
?	0006	
?	0007	
?	0008	
?	0009	
?	0010	
?	0011	

```
void main () {  
    int i;  
    i = 5;  
    int j = 'A';  
  
    int *x;  
    x = &j;  
  
    printf(„%d“, j);  
        $ 65  
    printf(„%d“, x);  
        $ 0001  
  
    printf(„%d“, *x);  
  
}
```

# Programmablauf:

Speicherbereich (beschreibbar)	Adressbereich (nur lesbar)	Namen
5	0000	i
65	0001	j
0001 Dereferenzierung	0002	x
?	0003	
?	0004	
?	0005	
?	0006	
?	0007	
?	0008	
?	0009	
?	0010	
?	0011	

```
void main () {  
    int i;  
    i = 5;  
    int j = 'A';  
  
    int *x;  
    x = &j;  
  
    printf(„%d“, j);  
        $ 65  
    printf(„%d“, x);  
        $ 0001  
  
    printf(„%d“, *x);  
  
}
```

# Programmablauf:

Speicherbereich (beschreibbar)	Adressbereich (nur lesbar)	Namen
5	0000	i
65	0001	j
0001 Dereferenzierung	0002	x
?	0003	
?	0004	
?	0005	
?	0006	
?	0007	
?	0008	
?	0009	
?	0010	
?	0011	

```

void main () {
    int i;
    i = 5;
    int j = 'A';

    int *x;
    x = &j;

    printf(„%d“, j);
        $ 65
    printf(„%d“, x);
        $ 0001

    printf(„%d“, *x);
        $ 65
}

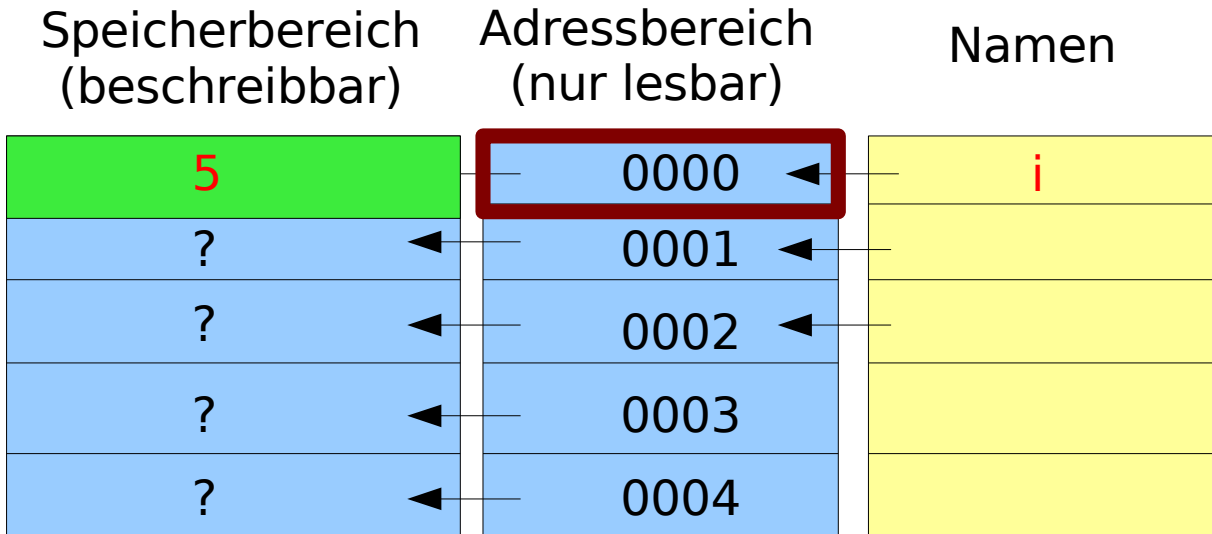
```



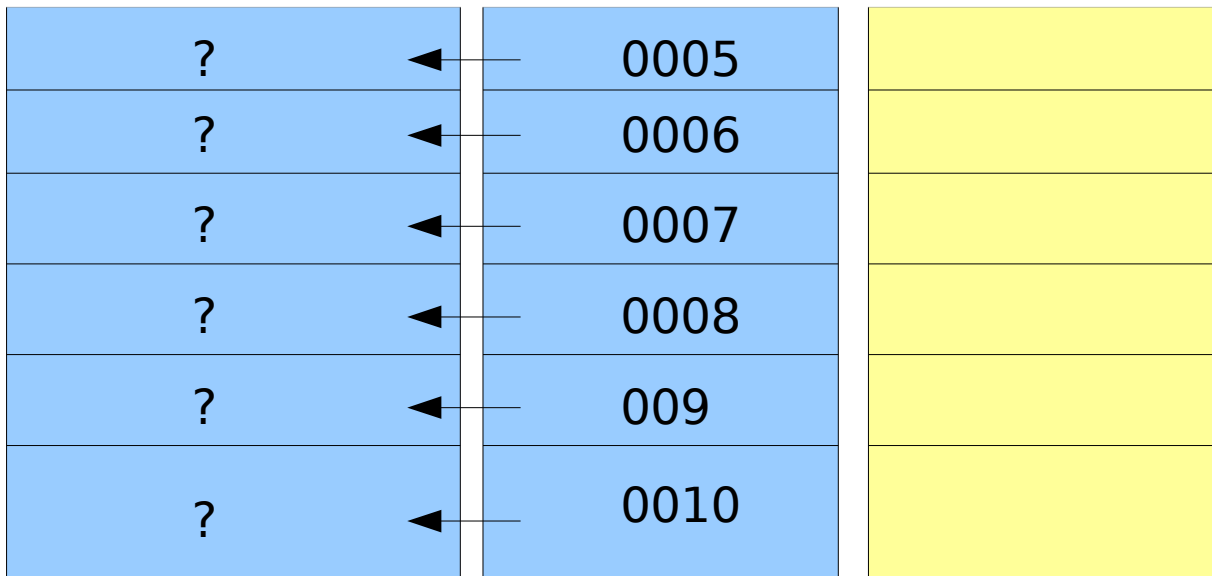
Pointer  
und  
Adressen  
bei

# **Funktionen**

# Programmablauf:



```
void out(int *p) {  
  
}
```



```
void main () {  
    int i = 5;  
  
}
```

# Programmablauf:

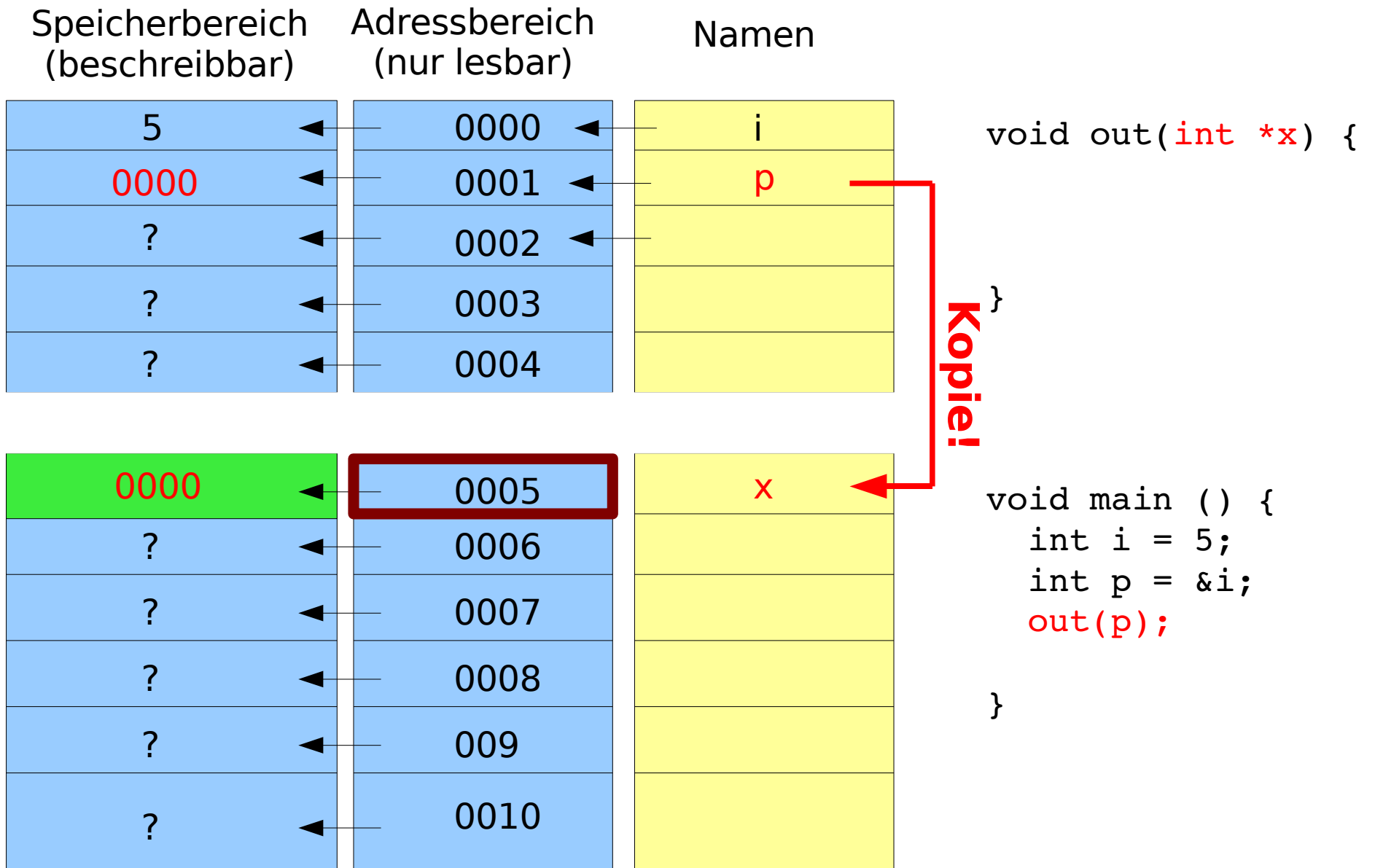
Speicherbereich (beschreibbar)	Adressbereich (nur lesbar)	Namen
5	0000	i
0000	0001	p
?	0002	
?	0003	
?	0004	

```
void out(int *p) {  
  
}
```

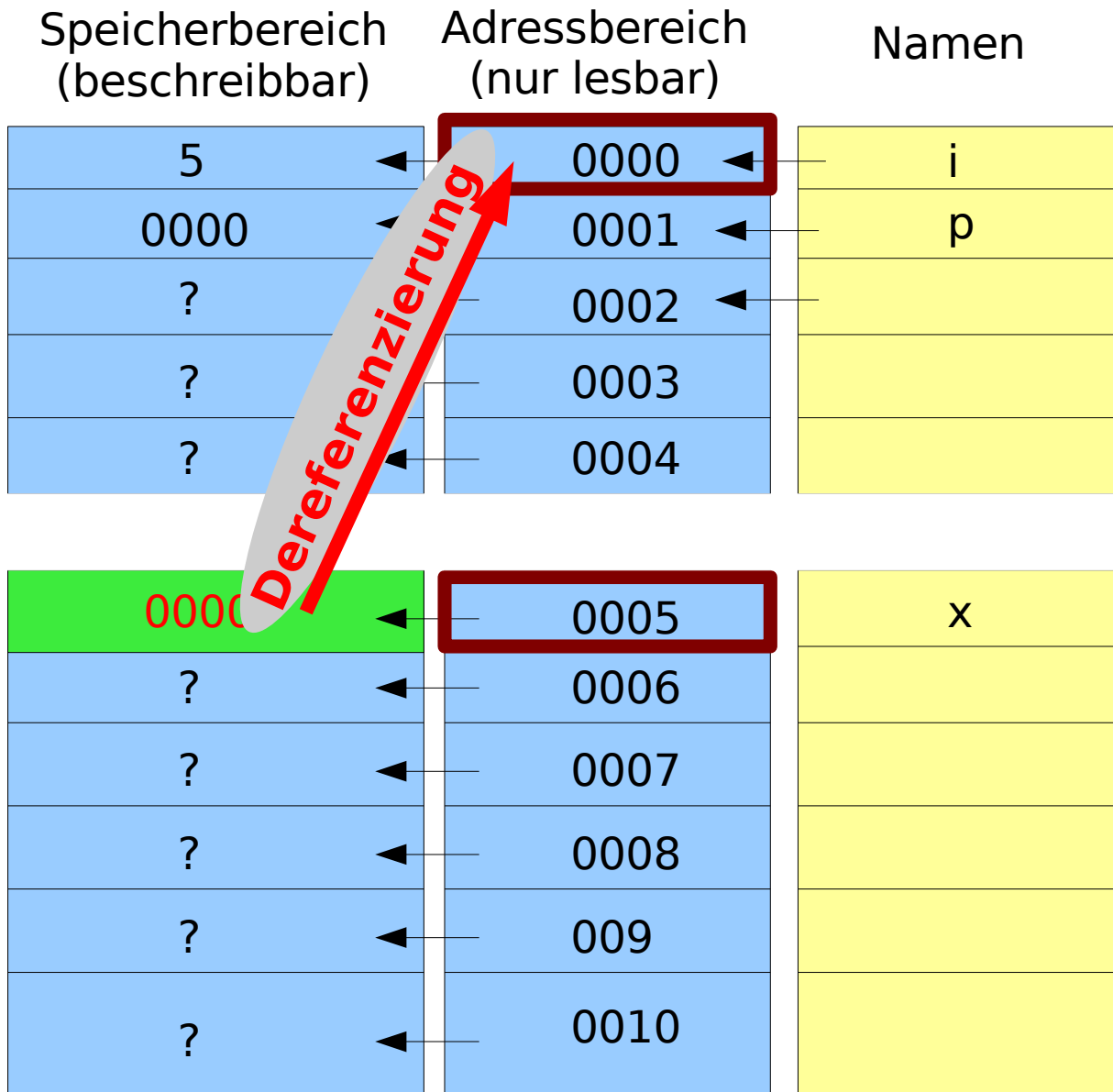
?	0005	
?	0006	
?	0007	
?	0008	
?	0009	
?	0010	

```
void main () {  
    int i = 5;  
    int p = &i;  
  
}
```

# Programmablauf:



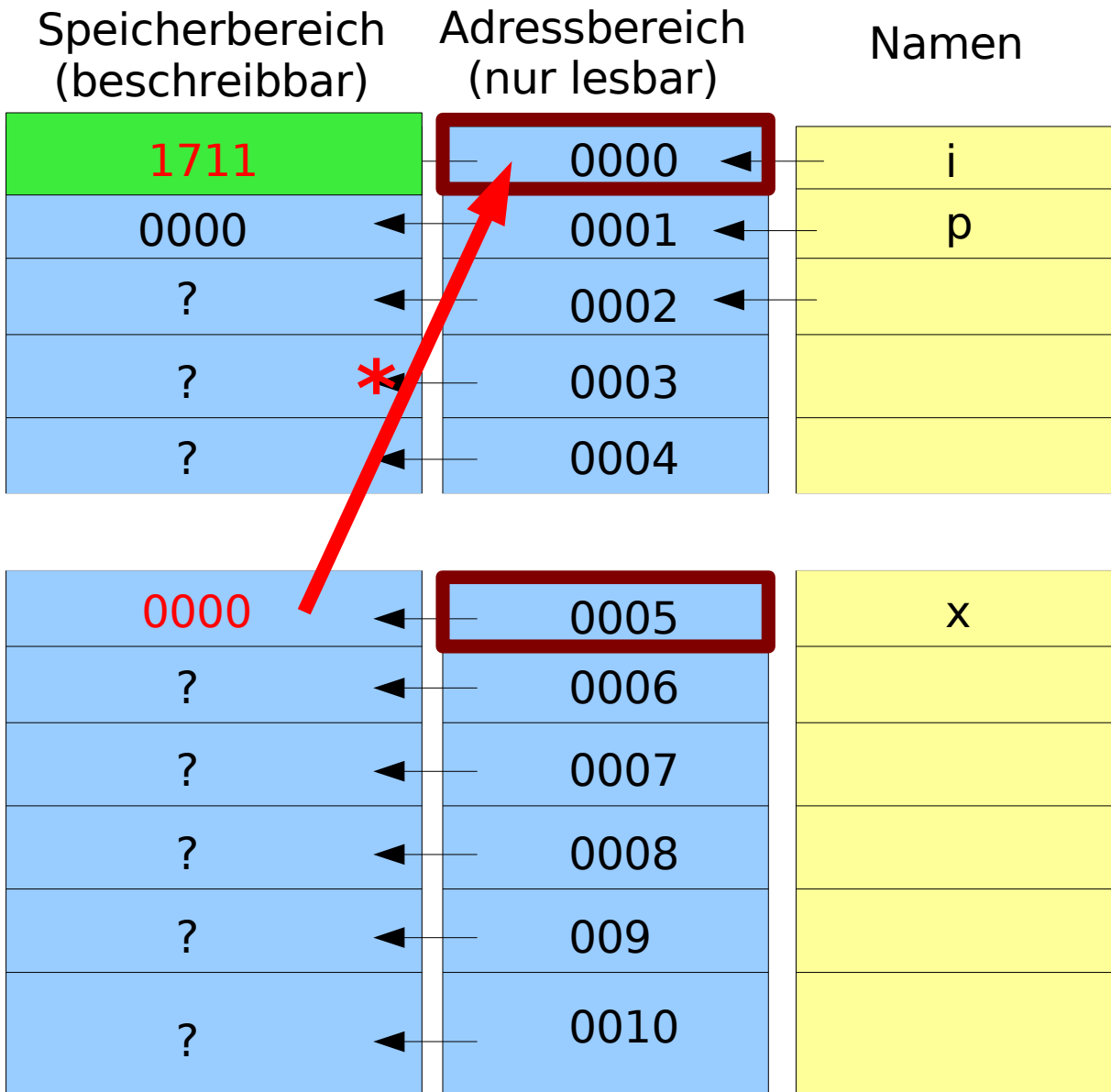
# Programmablauf:



```
void out(int *x) {  
    *x  
}
```

```
void main () {  
    int i = 5;  
    int p = &i;  
    out(p);  
}
```

# Programmablauf:



```
void out(int *x) {  
    *x = 1711;  
}
```

```
void main () {  
    int i = 5;  
    int p = &i;  
    out(p);  
}
```

# Programmablauf:

Speicherbereich (beschreibbar)	Adressbereich (nur lesbar)	Namen
1711 ←	0000 ←	i
0000 ←	0001 ←	p
? ←	0002 ←	
? ←	0003	
? ←	0004	

```
void out(int *x) {  
    *_x = 1711;  
}
```

**Variable zerstört!**

? ←	0005	
? ←	0006	
? ←	0007	
? ←	0008	
? ←	0009	
? ←	0010	

```
void main () {  
    int i = 5;  
    int p = &i;  
    out(p);  
}
```

# Programmablauf:

Speicherbereich (beschreibbar)	Adressbereich (nur lesbar)	Namen
1711 ←	0000 ←	i
0000 ←	0001 ←	p
? ←	0002 ←	
? ←	0003	
? ←	0004	

? ←	0005	
? ←	0006	
? ←	0007	
? ←	0008	
? ←	0009	
? ←	0010	

```
void out(int *x) {  
    *_x = 1711;  
}
```

```
void main () {  
    int i = 5;  
    int p = &i;  
    out(p);  
  
    printf(„%d“, i);  
    $ 1711  
}
```