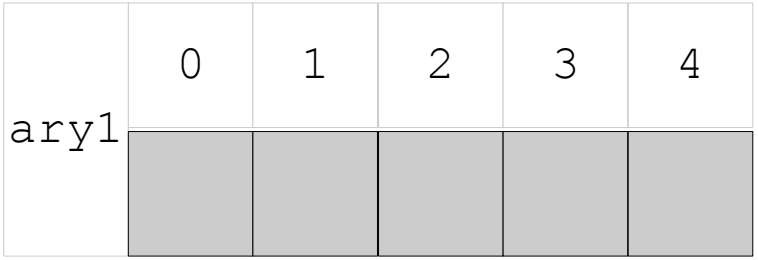
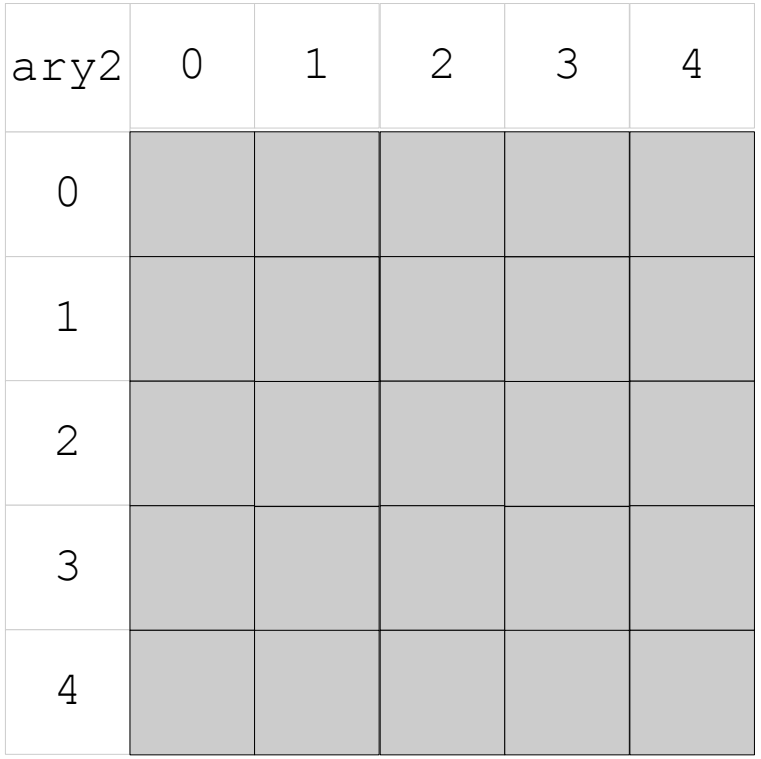
[PPI] 2. Masovne Instrukcije
2016/2017

Arrays

```
int ary1[5];

int ary2[5][5];
```





Arrays: Initialization

```
int ary1[5] = {1, 2, 3};
int ary2[5][5]; = {{1, 2, 3, 4, 5},
                   {6, 7, 8, 9, 10}};
```

ary1	0	1	2	3	4
	1	2	3	0	0

ary2	0	1	2	3	4
0	1	2	3	4	5
1	6	7	8	9	10
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

Arrays: Access

```
int ary1[5] = {1, 2, 3};
int ary2[5][5]; = {{1, 2, 3, 4, 5},
                   {6, 7, 8, 9, 10}};
```

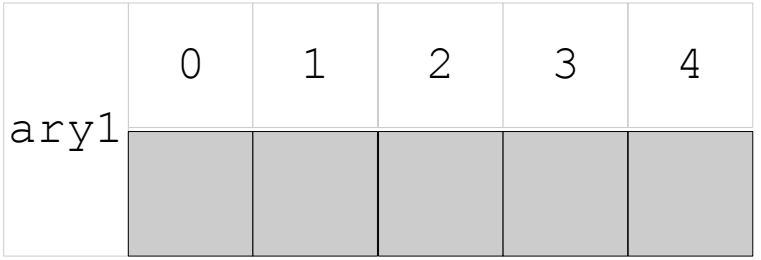
ary1	0	1	2	3	4
	1	2	3	0	5

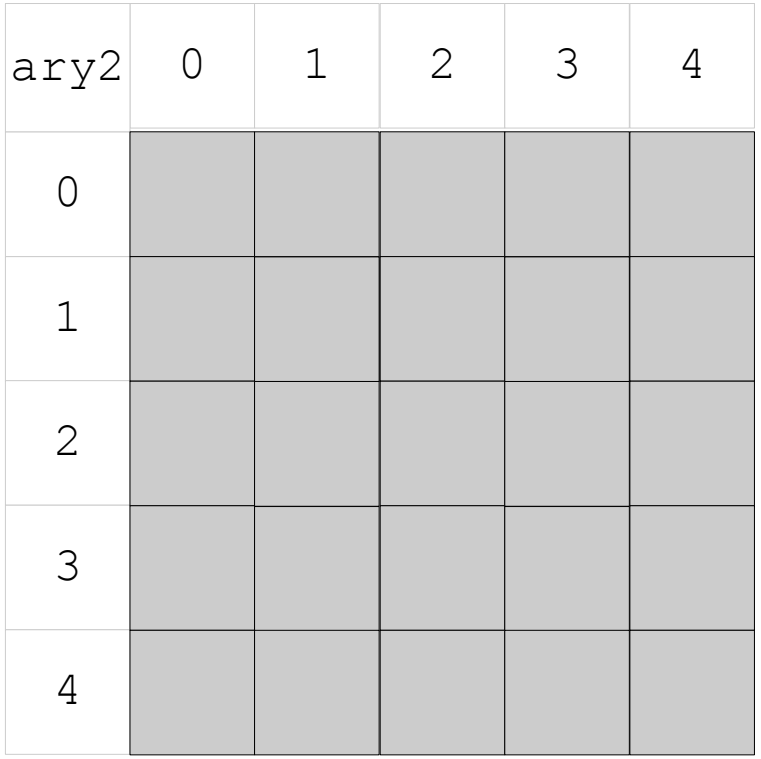
ary2	0	1	2	3	4
0	1	2	3	4	5
1	6	7	8	9	10
2	0	0	0	0	0
3	0	0	18	0	0
4	0	0	0	0	0

```
ary1[4] = 5;
ary2[3][3]; = 18;

printf("%d", ary1[3]);
   >> 0
printf("%d", ary2[2][1]);
   >> 8
```

Arrays: Processing

```
int ary1[5];
int ary2[5][5];
```



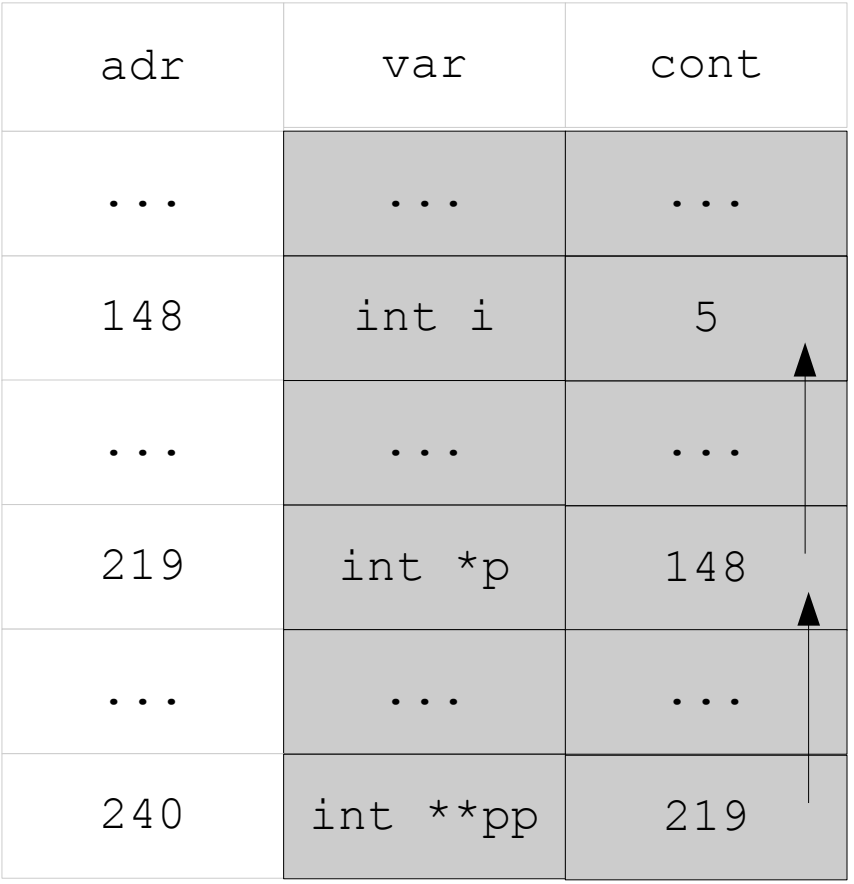



```
ary1[0]… ary1[1]…
ary1[2]… ary1[3]…
ary1[4]…

for (i = 0; i < 5, i++)
   ary1[i]…
```

```
ary1[0][0]… ary1[1][0]…
ary1[2][0]… ary1[3][0]…
ary1[4][0]… ary1[0][1]…

for (i = 0; i < 5, i++)
   for(j = 0; j< 5, j++)
      ary1[i][j]…
```

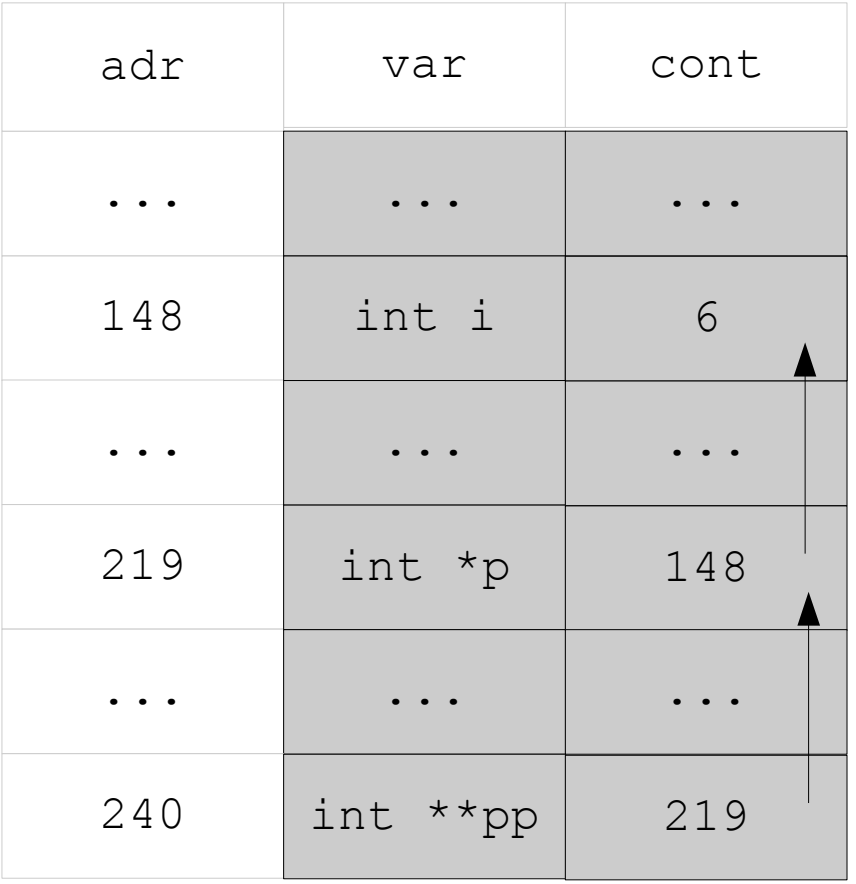
Pointers

```
int i;
int *p;
int **pp;


i = 5;
p = &i;
pp = &p;


printf("%d %d", *p, **p);
>> 5 5
```



Pointers

```
int i;
int *p;
int **pp;


*p = 6;


if (**p == 6)
  printf("%d %d", *p, **p)
>> 5 5
```



adr	var	cont
...	...	...
148	int i	6
...	...	...
219	int *p	148
...	...	...
240	int **pp	219

Pointers: Arithmetic & Arrays

```
int i[3]={10, 20, 30};
int *p;
int **pp;


p = &i[0]; //p = i;
printf("%d", *p,);
>> 10
printf("%d", *(p + 1));
>> 20
printf("%d", ++*(p + 2));
>> 31
```

adr	var	cont
110	int *i[3]	148
...	...	...
148	int[0]	10
152	int[1]	20
156	int[2]	31
...	...	...
219	int *p	148
...	...	...
240	int **pp	219

Pointers: Arithmetic & Arrays

```
int i[3]={10, 20, 30};
int *p;
int **pp;


p = &i[0]; //p = i;
printf("%d", *p,);
>> 10
printf("%d", *(p + 1));
>> 20
printf("%d", ++*(p + 2));
>> 31
```

adr	var	cont
110	int *i[3]	148
...	...	...
148	int[0]	10
152	int[1]	20
156	int[2]	31
...	...	...
219	int *p	148
...	...	...
240	int **pp	219

Functions

```
int func(int input) {...}
```

call by value vs call by reference