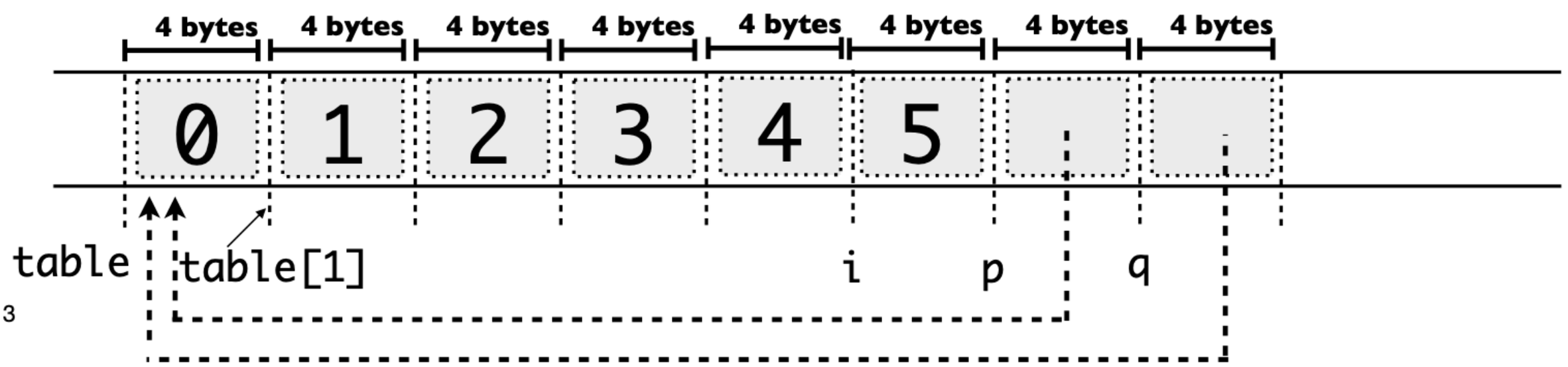
Pointers and Arrays

- Primitive arrays implemented in C using pointer to a block of continuous memory

```c
#define SIZE 5

int main(void)
{
   int table[SIZE];
   int i, *p, *q;

   /* fill up */
   for(i=0; i<SIZE; i++)
    table[i] = i;

   /* show content */
   for(i=0; i<SIZE; i++)
    printf("%d ", table[i]);
   printf("\n");

   p = &table[0];
   q = table;
   /* continues next slide */
```

sets p and q to element 0 of table



Pointers en Arrays

Als functieparameters:

```
int f(int array[]) {
 . . .
}
int a[] = {1, 2, 3};
f(a);
```

is identiek aan

```
int f(int *array) {
 . . .
}
int a[] = {1, 2, 3};
f(a);
```

Want “a” evalueert toch naar een adres

Als globale en lokale variabelen:

```
int a[] = {1, 2, 3};
```

“Maak een array van
3 integers aan”

is **NIET** identiek aan

```
int *a = {1, 2, 3};
```

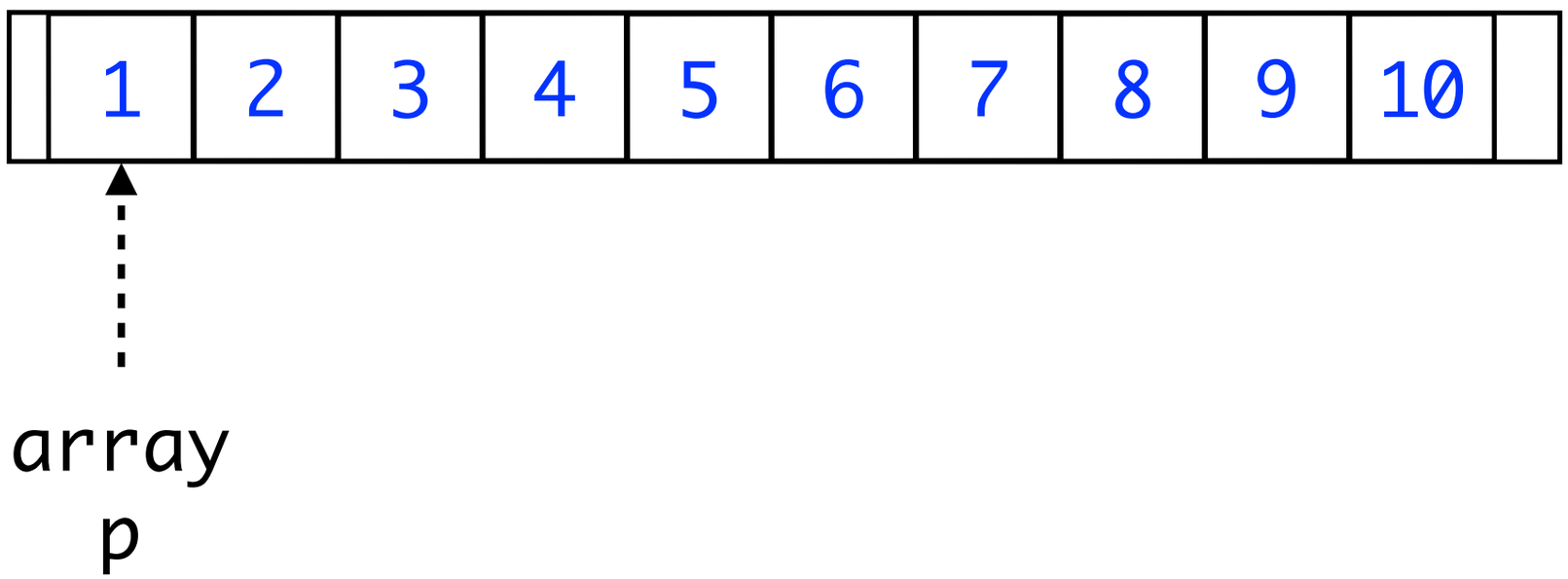
Compiler error

Pointer Arithmetic

```
int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int *p = array; // *p = &(array[0]);
for (int i = 0; i < 10; i++) {
  printf("%i", *(p + i));
}
```

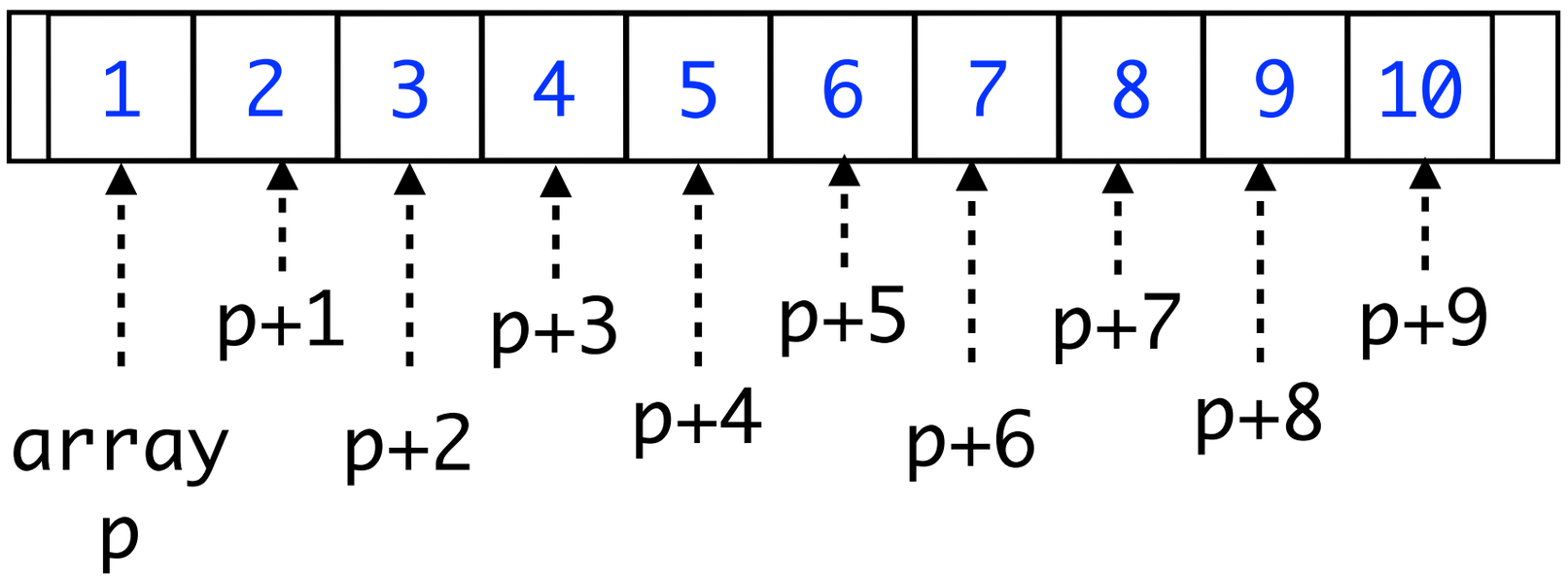
Pointer Arithmetic

```
int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int *p = array;
for (int i = 0; i < 10; i++) {
  printf("%i", *(p + i));
}
```



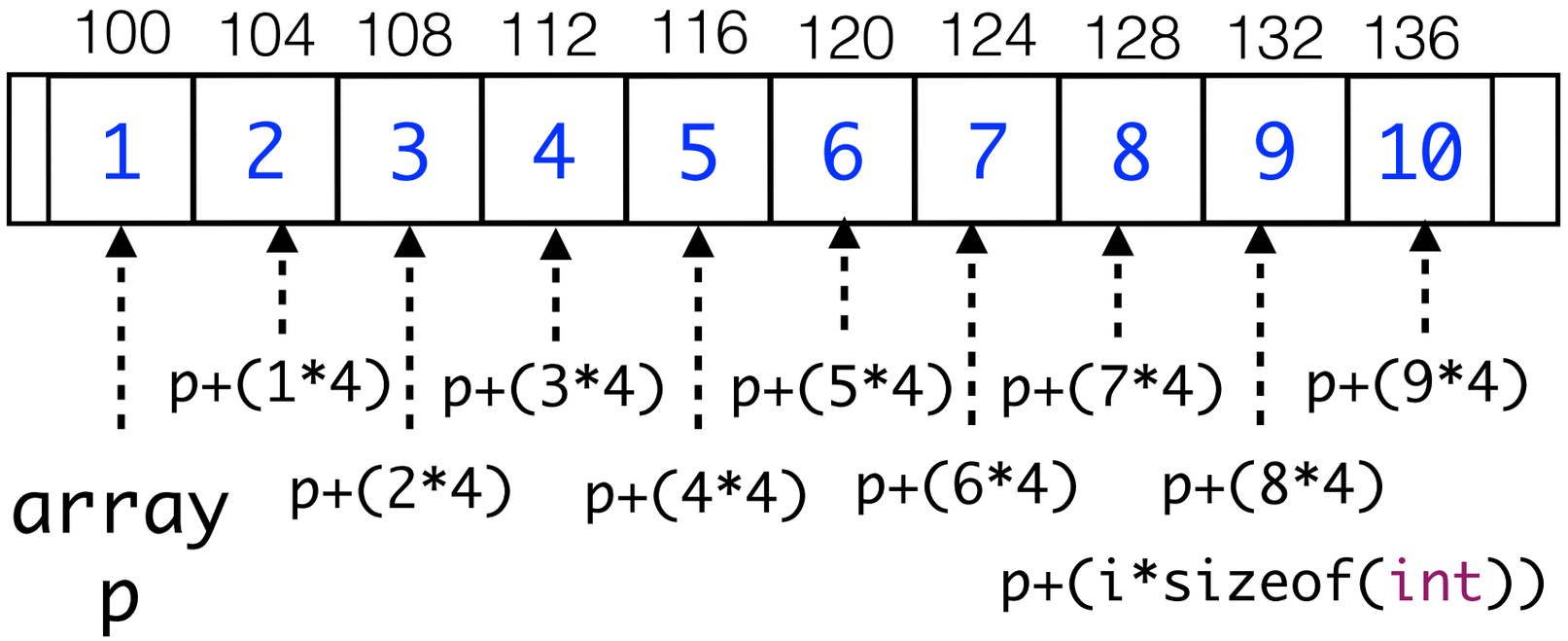
Pointer Arithmetic

```
int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int *p = array;
for (int i = 0; i < 10; i++) {
  printf("%i", *(p + i));
}
```



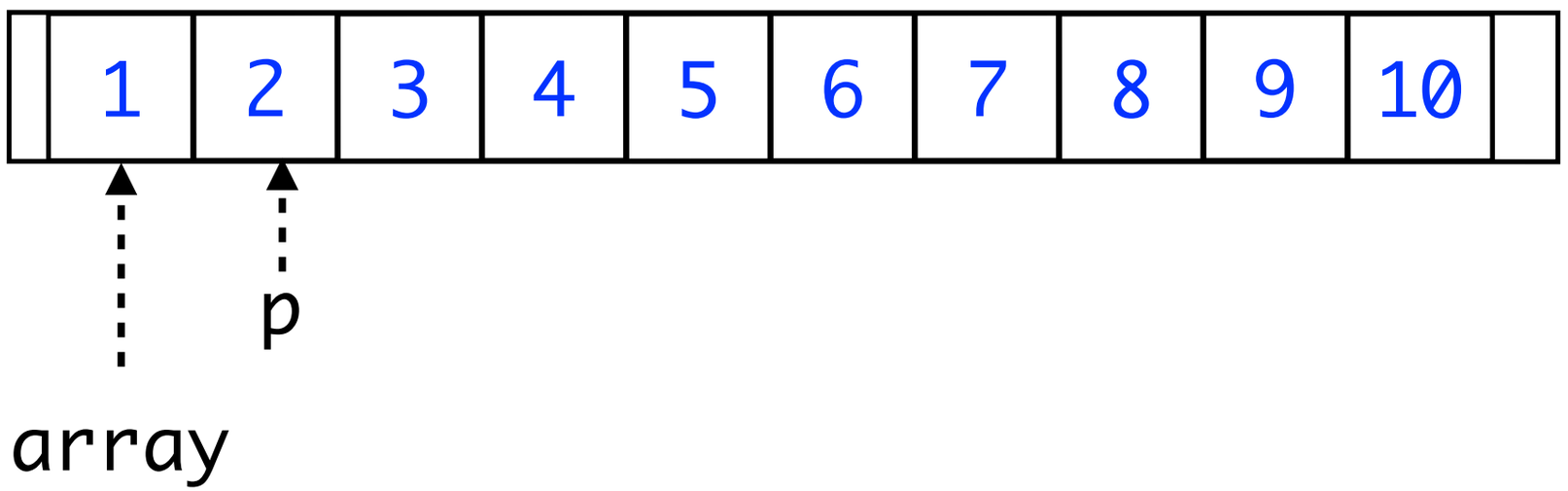
Pointer Arithmetic

```
int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int *p = array;
for (int i = 0; i < 10; i++) {
  printf("%i", *(p + i));
}
```



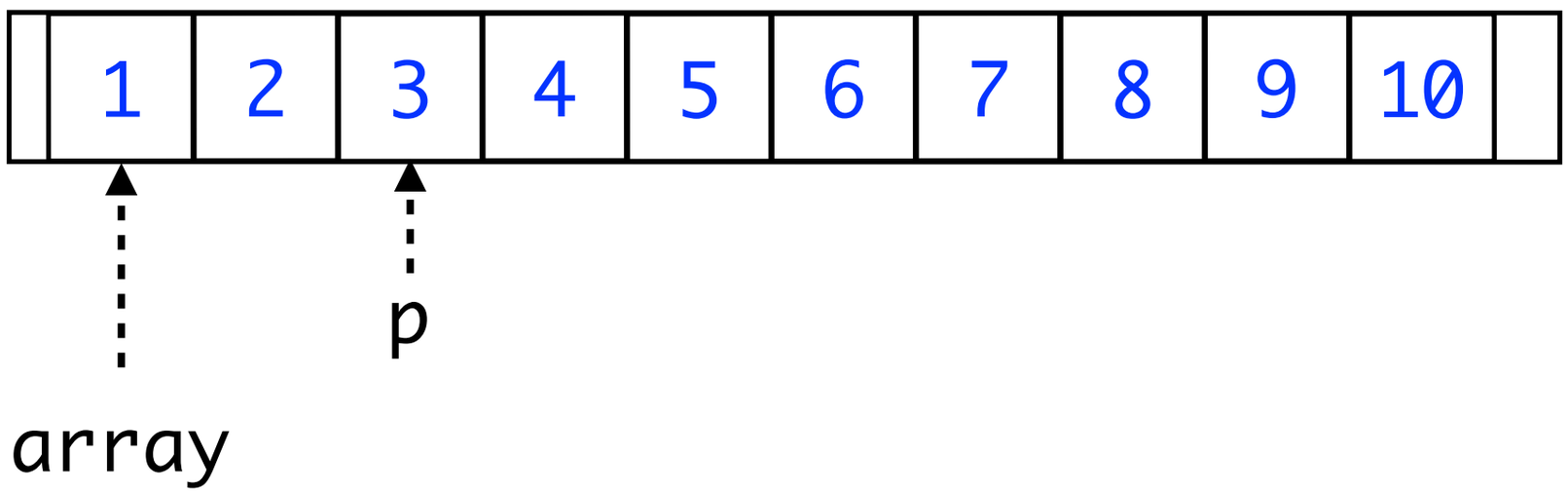
Pointer Arithmetic

```
int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int *p = array;
for (int i = 0; i < 10; i++) {
  printf("%i", *p);
  p++;
}
```



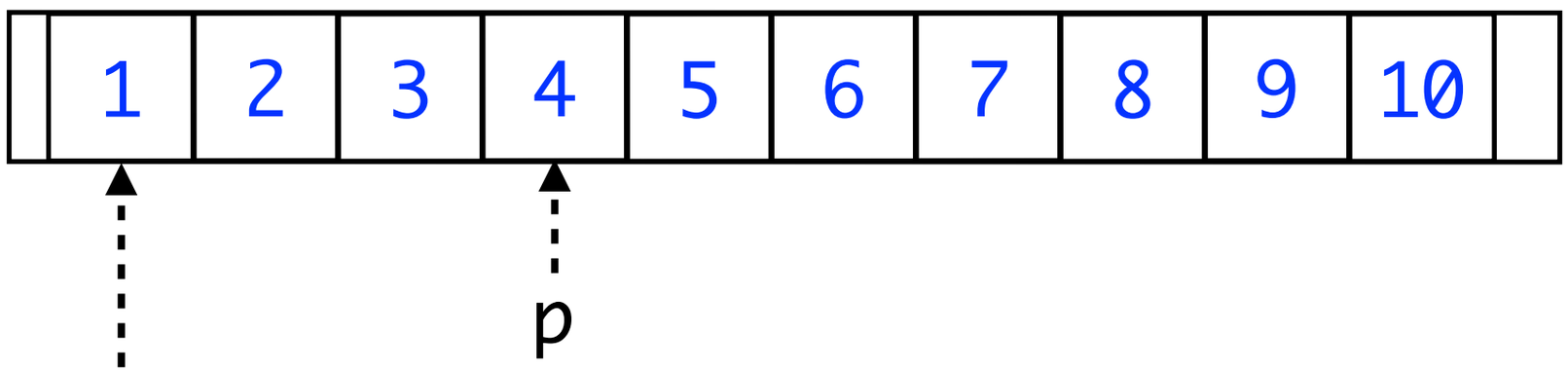
Pointer Arithmetic

```
int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int *p = array;
for (int i = 0; i < 10; i++) {
  printf("%i", *p);
  p++;
}
```



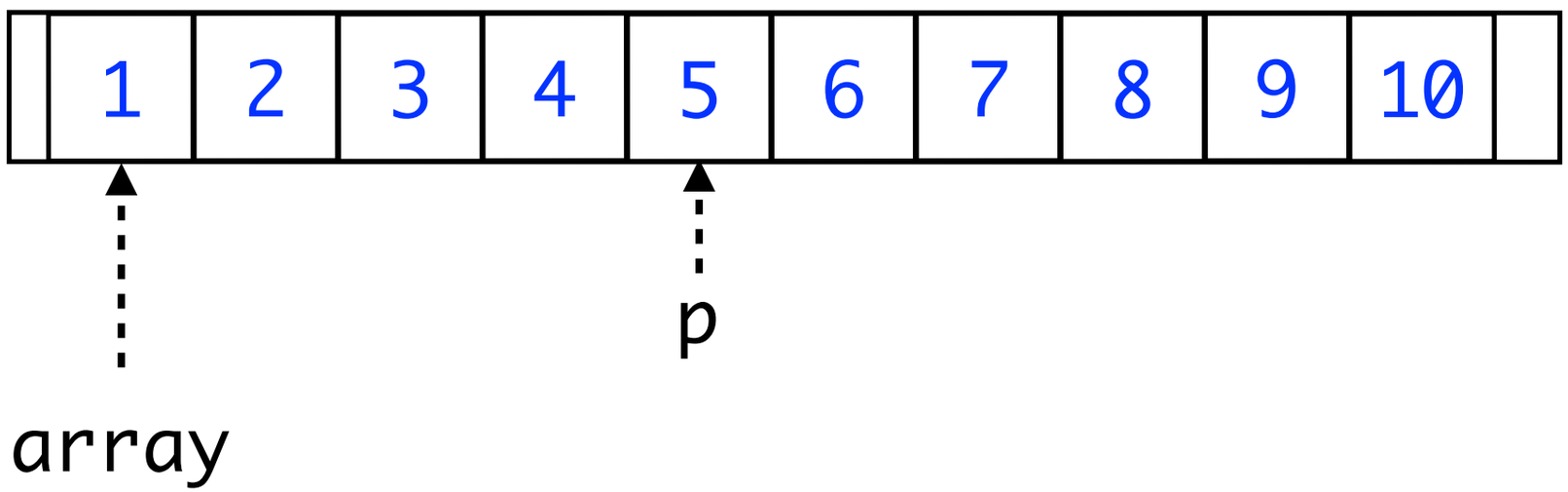
Pointer Arithmetic

```
int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int *p = array;
for (int i = 0; i < 10; i++) {
  printf("%i", *p);
  p++;
}
```



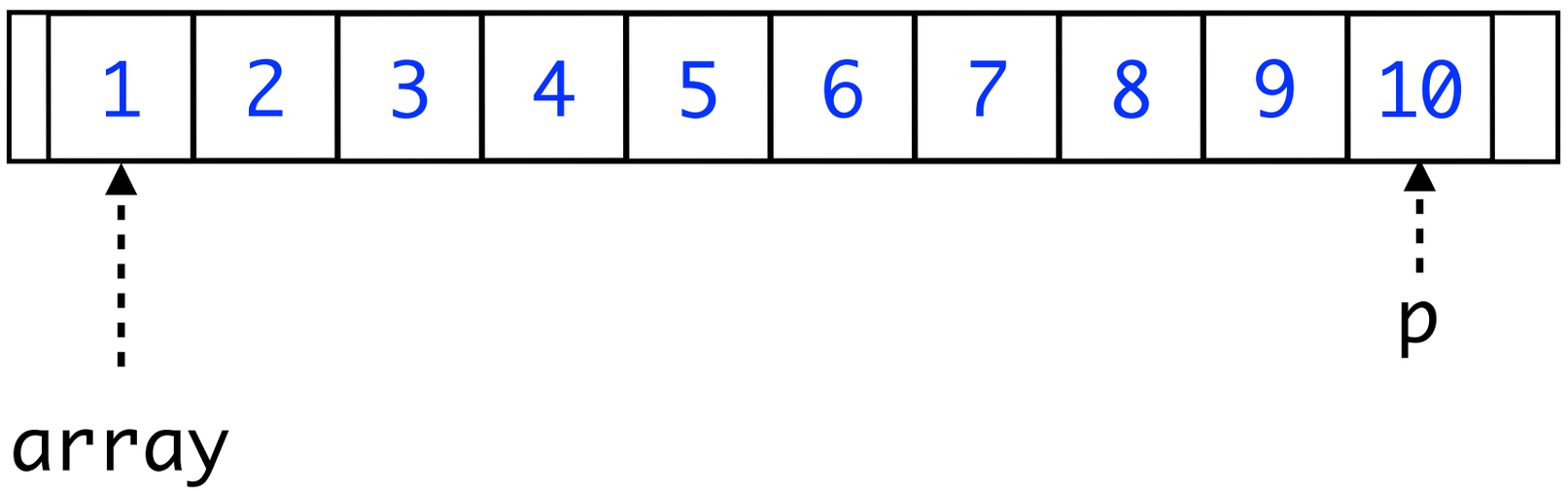
Pointer Arithmetic

```
int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int *p = array;
for (int i = 0; i < 10; i++) {
  printf("%i", *p);
  p++;
}
```



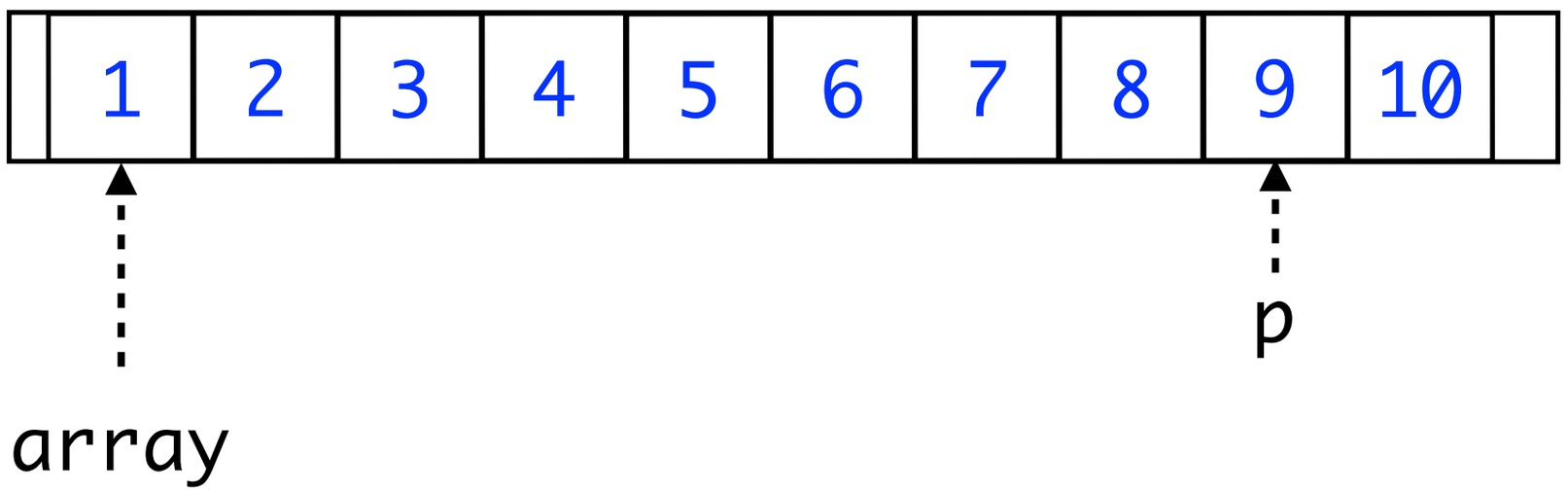
Pointer Arithmetic

```
int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int *p = array + 9;
for (int i = 0; i < 10; i++) {
  printf("%i", *p);
  p--;
}
```



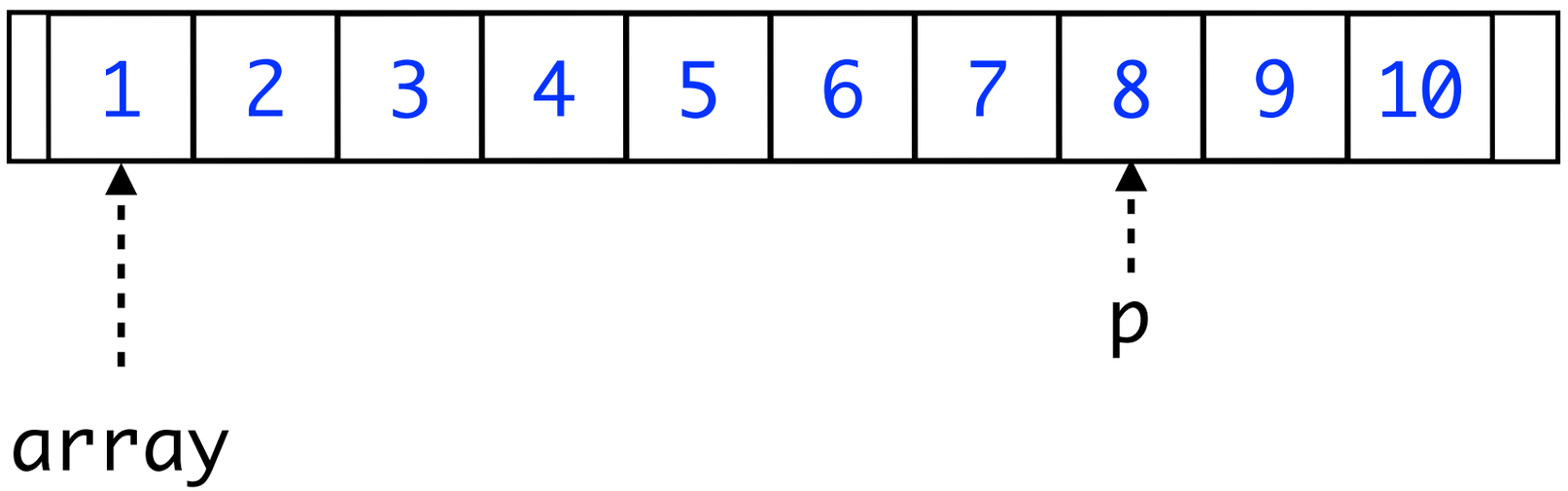
Pointer Arithmetic

```
int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int *p = array + 9;
for (int i = 0; i < 10; i++) {
  printf("%i", *p);
  p--;
}
```



Pointer Arithmetic

```
int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int *p = array + 9;
for (int i = 0; i < 10; i++) {
  printf("%i", *p);
  p--;
}
```



Samenvatting

```
&array[i] == p+i

 array[2] == *(array+2)

array[i] == *(p+i) == p[i] == *(array + i)

array > p   array == p   p <= array
```