

Lab1 Key points

- `&var` : μας φέρνει τη διεύθυνση που είναι αποθηκευμένη η `var`
- `int *p=&var` : δημιουργεί μεταβλητή τύπου `pointer` (δηλαδή ικανή να αποθηκεύσει διεύθυνση)
- `*p`: παίρνετε το περιεχόμενο της διεύθυνσης που έχει αποθηκεύσει ο `pointer`
- `p`: παίρνετε το περιεχόμενο του `pointer` (δηλαδή την τιμή της διεύθυνσης που είναι η `var`)
- `a[i]=*(a+i)`
 - `a`: η διεύθυνση αρχής του πίνακα, `+i` τόσες θέσεις μετά (πραγματικές θέσεις ανάλογα με τον τύπο των στοιχείων του πίνακα)
- `&a[i]`: διεύθυνση του `i` στοιχείου
- `*p++` : επιστρέφει το τωρινό περιεχόμενο της μεταβλητής που δείχνει ο `pointer` και αυξάνει την τιμή του ίδιου του `pointer` (επόμενη διεύθυνση)

Lab1

- Διαφορά array και &array
- Το array+1 σας πάει στην αμέσως επόμενη θέση μνήμης με βάση το τύπο του πίνακα (char +1, int +4 κλπ) σε σχέση με την αρχή του πίνακα, δηλαδή
 - αρχή πίνακα+μέγεθος τύπου
- Το &array σας πάει 1 θέση μετά το τέλος όλου του πίνακα, δηλαδή
 - αρχή πίνακα+θέσεις πίνακα*μέγεθος τύπου.

```
//example of difference between &array and array[]
int array[5];

printf("array=%p : &array=%p\n", array, &array);
int i=2;
printf("array+1 = %p : &array + 1 = %p, &array[i]+1 = %p\n", array+1, &array+1, &array[i]+1);
```

```
check value of array (p) 0x7ffdf8af0db0
array=0x7ffdf8af0db0 : &array=0x7ffdf8af0db0
array+1 = 0x7ffdf8af0db4 : &array + 1 = 0x7ffdf8af0dc4, &array[i]+1 =0x7ffdf8af0dbc
```