

local → ตัวแปรที่ถูกประกาศ
ในฟังก์ชัน รวมทั้งตัวที่เป็น parameter ด้วย

automatic storage duration → เหมือน local (ไม่คง)
static " " → " global (คงอยู่)

Array

→ ตัวแปร 1 ตัว สามารถเก็บค่าได้หลายค่า แต่ต้องเป็นประเภทเดียวกัน

ค่าคงที่ { #define ตัวแปร ค่า → global
const type ตัวแปร = ค่า; → local

int A[2] = { 10, 11, 12 }

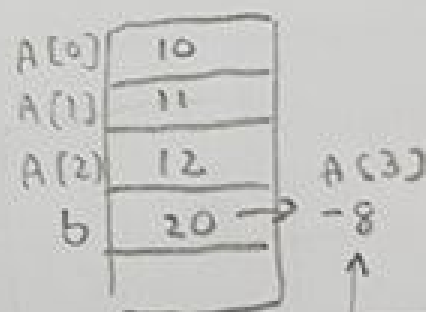
int b = 20

A[3] = -4

printf("%d\n", A[1]);

" " , A[3]);

" " b



buffer overflow
(เมื่อใช้ Array เกินที่ที่กำหนดขึ้นไว้)

11
-8
-8

lab 4

- `printf("reference of x %lu", &x);` → \rightarrow Address of x
- `printf(" " " " %p", &x);` → " " " " 16
- \rightarrow Array \rightarrow Index 0 address number = $\text{number}[0]$
- $\text{Array} + (\text{sizeof type}) \rightarrow$ address of array[0] + unit byte [16]
 \rightarrow `%p` of number + 1 → address number + 4
- Pointer → \rightarrow address (& array, Array)
`type *arrptr;` `type *arrptr = arr;`
`arrptr = arr;`
`int num[] = { 1, 2, 3 };`
`int *arrptr;`
`arrptr = num;` \rightarrow `arrptr = address of num[0]` \rightarrow +4
`arrptr += 1;` \rightarrow `arrptr = " " [1]` (16)
`*arrptr += 3;` \rightarrow `num[1] + 3` \rightarrow `num = { 1, 5, 3 }`
- `scanf`

`char S[100];`

`int i;`

`char c;`

`scanf("%d=%c=%s", &i, &c, S);`

đọc dữ liệu vào format của biến để lưu trữ

Array là mảng lưu trữ

Segmentation

`char *c;`

`int *i;`

`float *f;`

lab 5

- Value by

- pointer

`void`

`for(i`

`pr`

`*`

`}`

`}`

//

- Linked list

- structure

`struct`

`}`

- `msl typ`