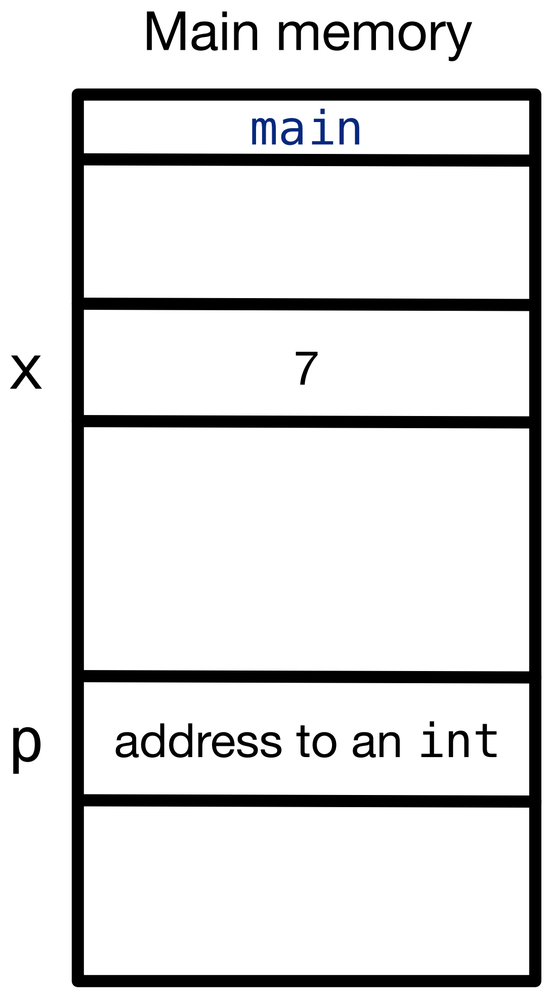
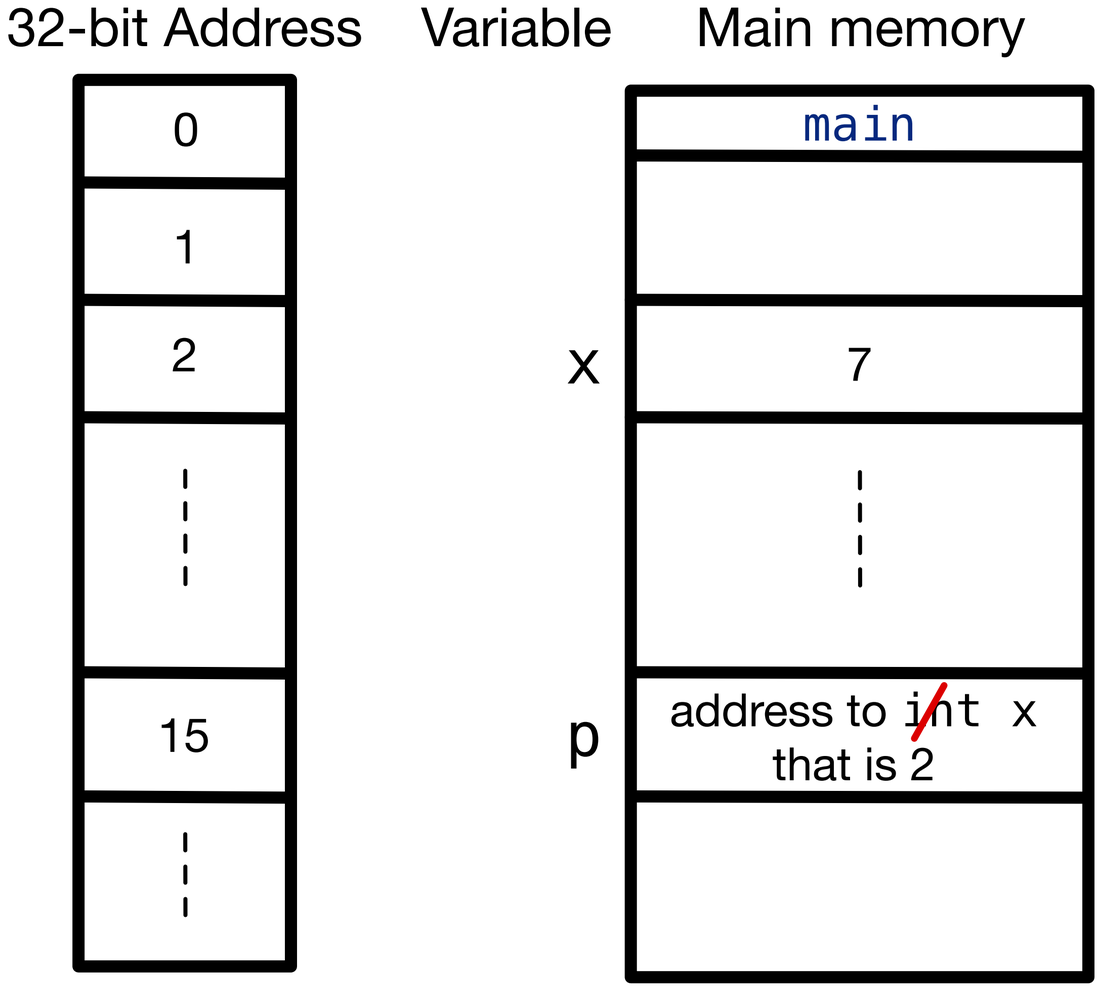



```c
#include <stdio.h>

int main(void) {
    int x = 7;
    int *p;

    return 0;
}
```

① `int x = 7;` data type of x is `int`

② `int *p;` data type of p is `int*`



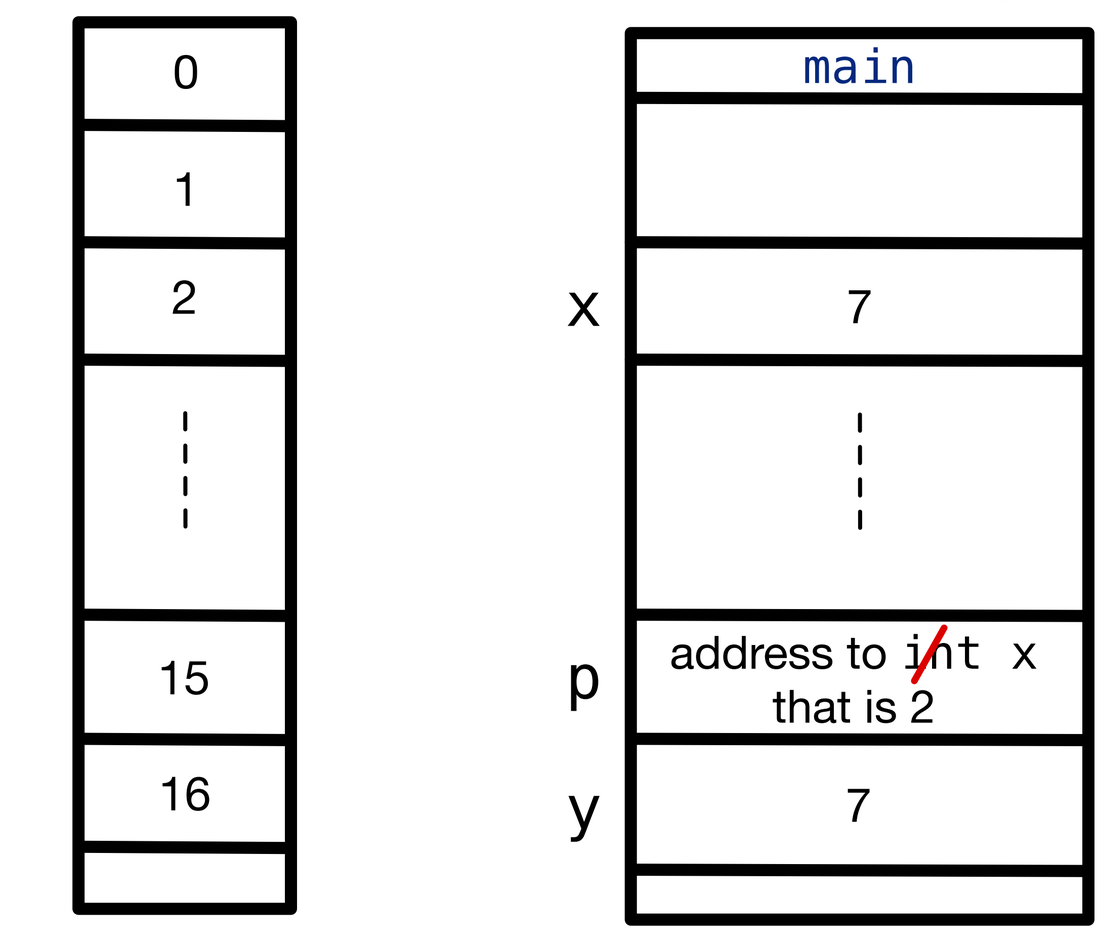

```c
#include <stdio.h>

int main(void) {
    int x = 7;
    int *p;
    p = &x;

    return 0;
}
```

① `int x = 7;` data type of x is `int`

② `int *p;` data type of p is `int*`

③ `p = &x;` p is assigned the address of x

Reference operator: this means “address of”
Recall its usage in `scanf`




```c
#include <stdio.h>

int main(void) {
    int x = 7;
    int *p;
    p = &x;
    int y;
    y = *p;

    return 0;
}
```

① `int x = 7;` data type of x is `int`

② `int *p;` data type of p is `int*`

③ `p = &x;` p is assigned the address of x

④ `int y;` declare variable y

⑤ `y = *p;` y is assigned the value of the variable stored in address p that is x

Dereference operator: this means “value at” address p