pointer : variable which stores address
↓
the content is address
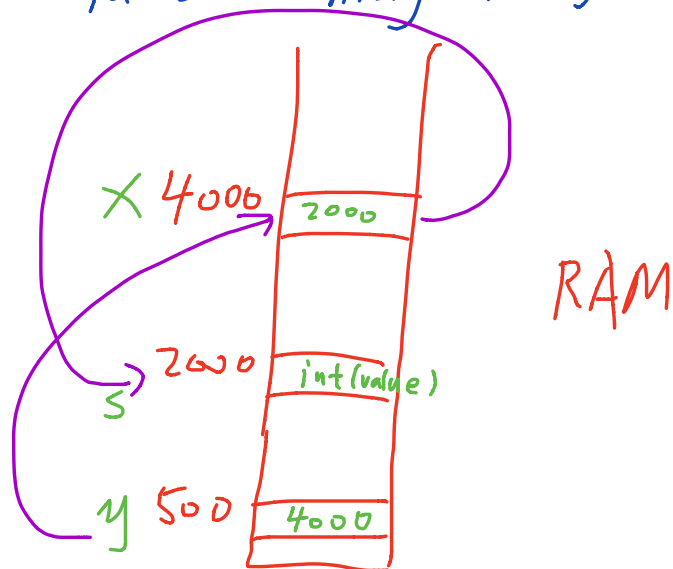
X 7000 | 1000

S 1000 | 50
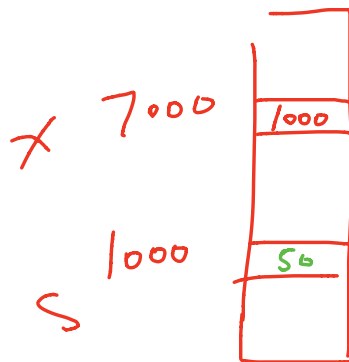
int * x,

x = 1000

*x → 50

---

main( ){

int S;

int *x;

int **y;

variable is just address
in a running program

X 4000 | 2000

S 2000 | int (value)

y 500 | 4000

RAM

Indirection

```
int x;
int *y;
int **z;

    x = 5;
    printf("%d\n",x);
    printf("%p\n",&x);

    y = &x;
    printf("%d\n",*y);
    printf("%p\n",y);

    z = &y;
    printf("%d\n",**z);
    printf("%p\n",z);

}
```

```
// C++ is a superset of C with OO (objective oriented) extensions.
// However, there are subtle differences that must be taken into account
// when porting C code.

#include <stdio.h>

int main()
{
int x;
void changeval(int &);

    x = 5;
    printf("%d\n",x);
    changeval(x);
    printf("%d\n",x);

}

void changeval(int &a)
{
    a = 3;
}
```

"main.c" 23L, 335C

int &
pass by address
reference

changeval(X)

```c
// Basic, rudimentary way to track down where in code a run-time error
// occurs. Use fflush() to ensure buffered output is flushed.

#include <stdio.h>

#define MYDEBUG

int main()
{
int x;
void changeval(int *);

  x = 5;
  printf("%d\n",x);

#ifdef MYDEBUG
printf("ok 1\n");
#endif

  changeval(&x);

#ifdef MYDEBUG
printf("ok 2\n");
#endif

  printf("%d\n",x);
  changeval(x);

#ifdef MYDEBUG
printf("ok 3\n");
#endif
```

(Handwritten annotations:)

// #define MYDEBUG
commen this line

all these will be removed

Bug

&x
should pass address
changeval(&x);

segmentation Fault

addvess 5

is not belong
to your a.out

1500
1000
5

% a.out
load a.out