

Overview (1A)

Copyright (c) 2009 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice.

Calculating the Mean

The mean of 3 numbers

$$m = \frac{a + b + c}{3}$$

$$\frac{40 + 50 + 60}{3} = 50 \quad \text{Integer number}$$

$$\frac{45 + 53 + 63}{3} = \frac{161}{3} = 53.6666666... \quad \text{Real number}$$

Calculating the Mean - in C

```
int    a, b, c;  
int    mean;
```

```
a = 40;
```

```
b = 50;
```

```
c = 60;
```

```
mean = (a + b + c) / 3;
```

```
int    a, b, c;  
float  mean;
```

```
a = 45;
```

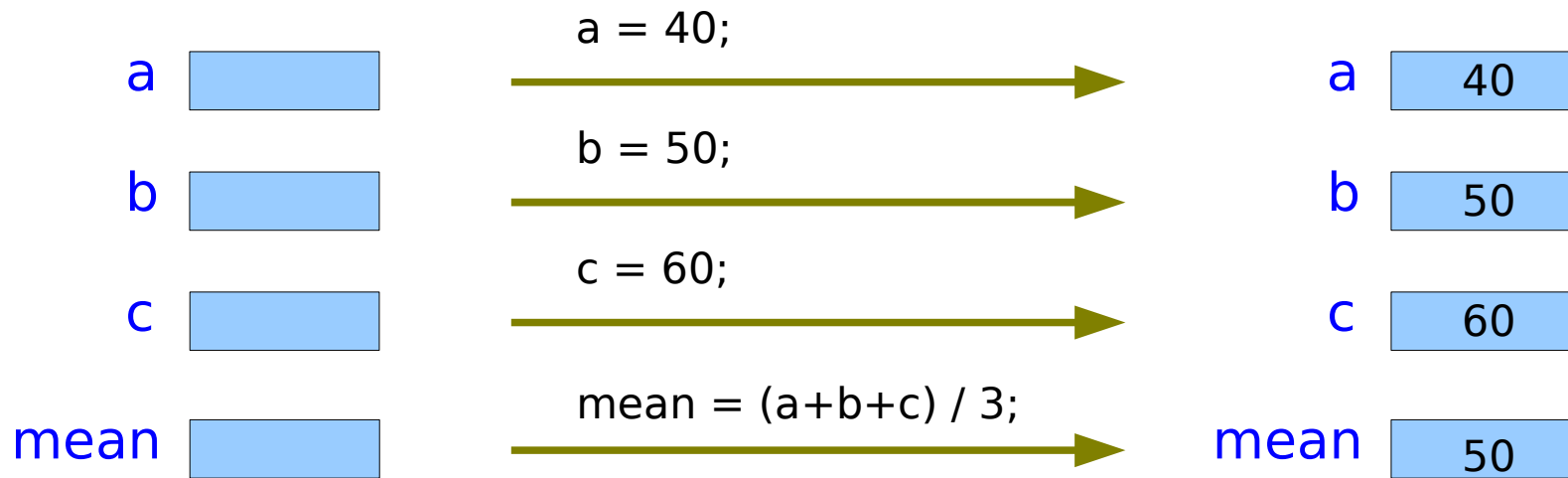
```
b = 53;
```

```
c = 63;
```

```
mean = (a + b + c) / 3.0;
```

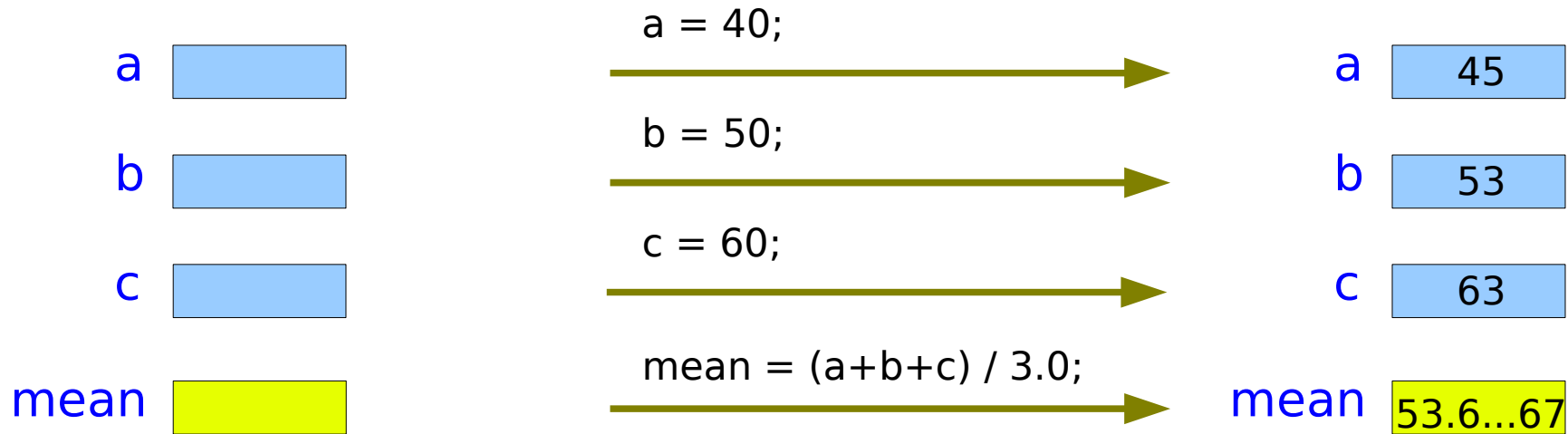
- * Variable
- * Type
- * Assignment
- * Operator

Variable - Int



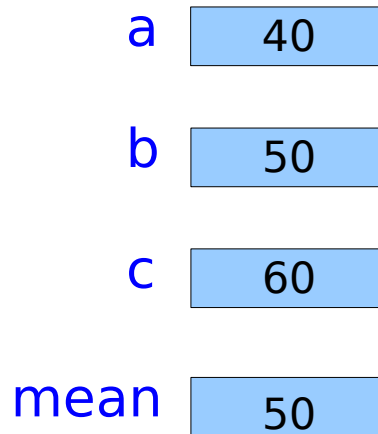
int type

Variable - Float



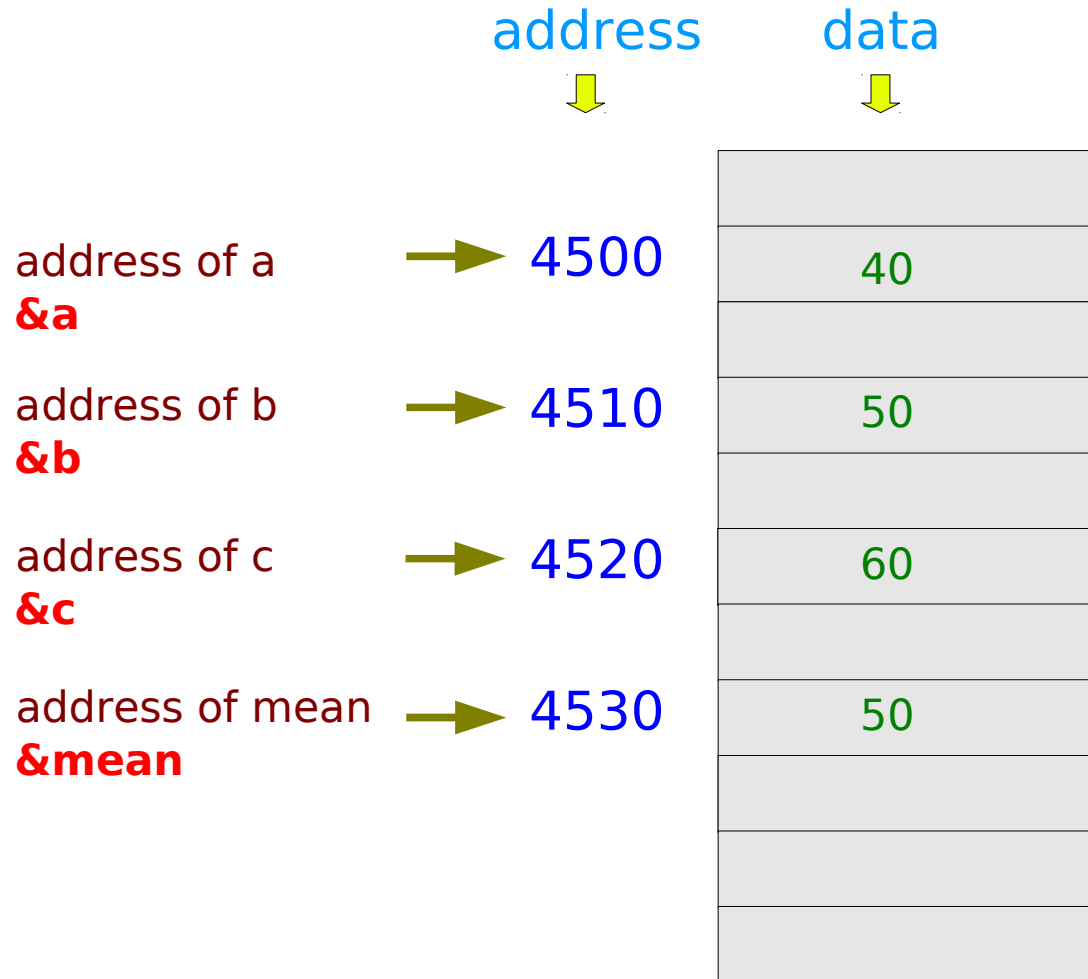
float type

Memory and & operator

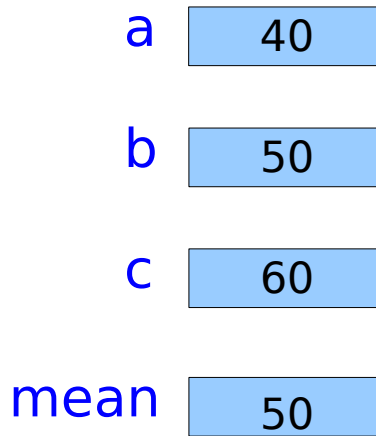


`a = 40;`

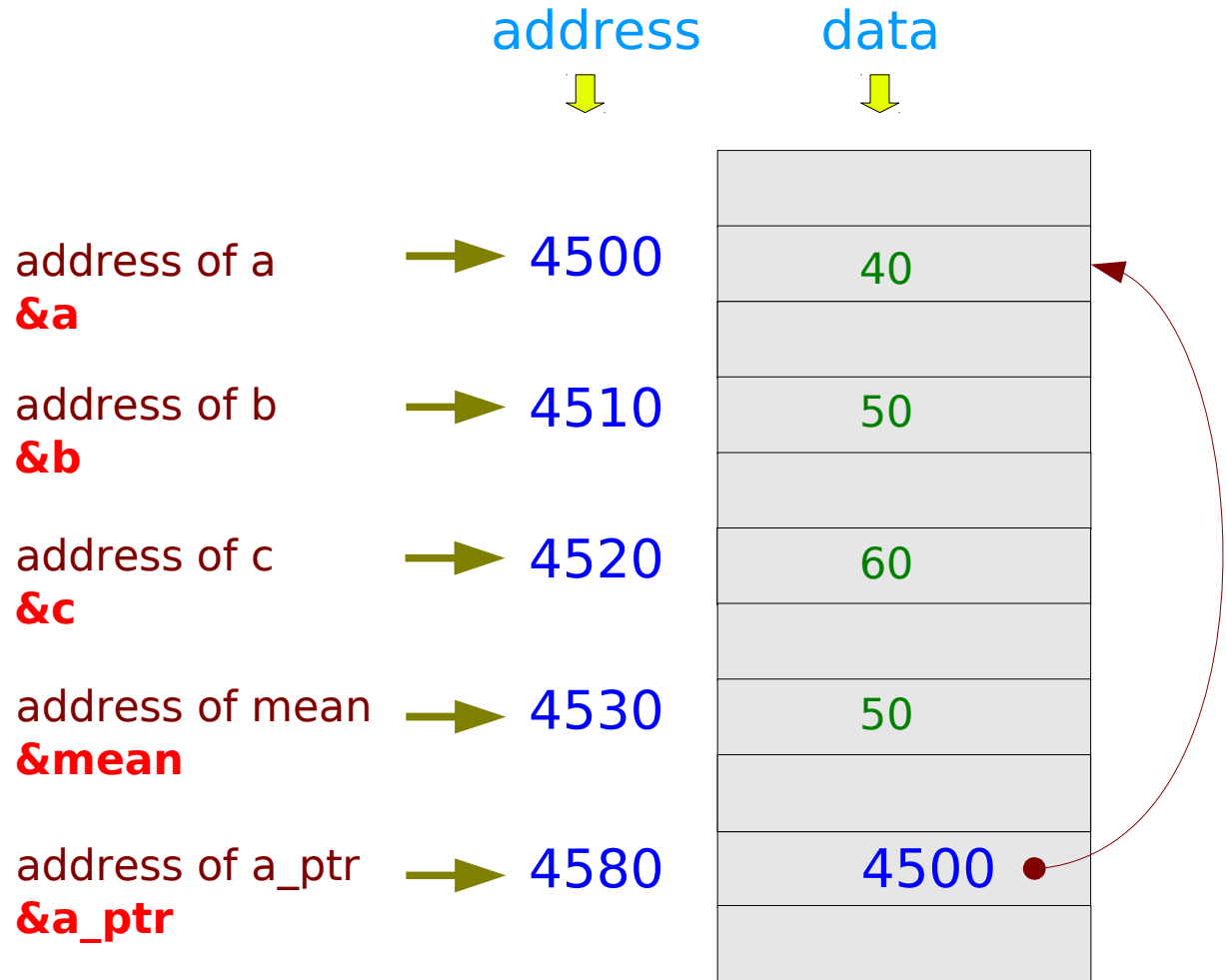
where: address of a
what: data 40



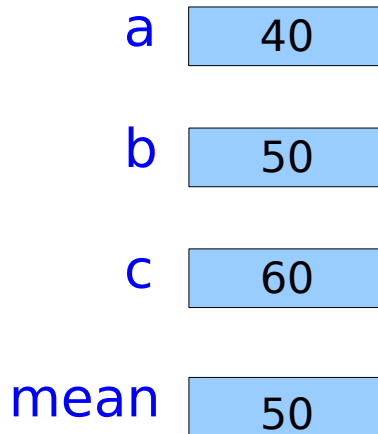
Memory and Pointer



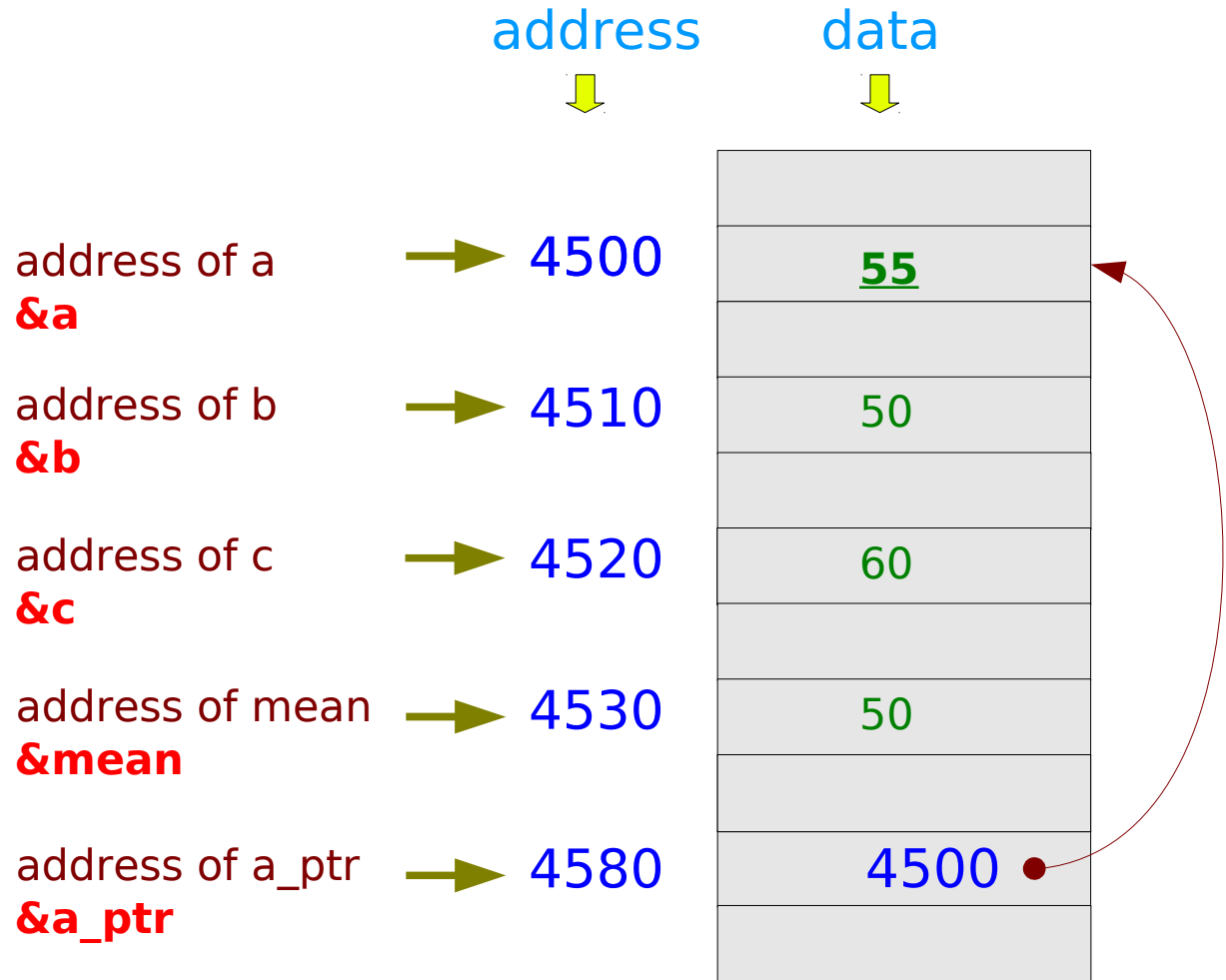
`a_ptr = &a;`
where: addr of a_ptr
what: data &a
(addr of a)



Memory and * operator



`*a_ptr = 55;`
where: data of `a_ptr`
 (→ another addr)
what: data 55



Pointer Type

```
int a;
```

`a` holds data

`&a`

value of `a`

```
int *a_ptr;
```

`a_ptr` holds address
`*a_ptr` holds data

`&a_ptr`

value of `a_ptr`

value of `a_ptr`
(→ an address)

`*a_ptr`

Printf Function

Expected Output

The mean of three numbers

a = 40

b = 50

c = 60

mean(40, 50, 60) => 50

```
printf("The mean of three numbers \n");  
printf("a = %d \n", a);  
printf("b = %d \n", b);  
printf("c = %d \n", c);  
printf("mean (%d, %d, %d) => %d \n", a, b, c, mean);
```

Main Function (1)

```
main (void)
{
    int    a, b, c;
    int    mean;

    a = 40;
    b = 50;
    c = 60;

    mean = (a + b + c) / 3;

    printf("The mean of three numbers \n");
    printf("a = %d \n", a);
    printf("b = %d \n", b);
    printf("c = %d \n", c);
    printf("mean (%d, %d, %d) => %d \n", a, b, c, mean);

}
```

Scanf Function

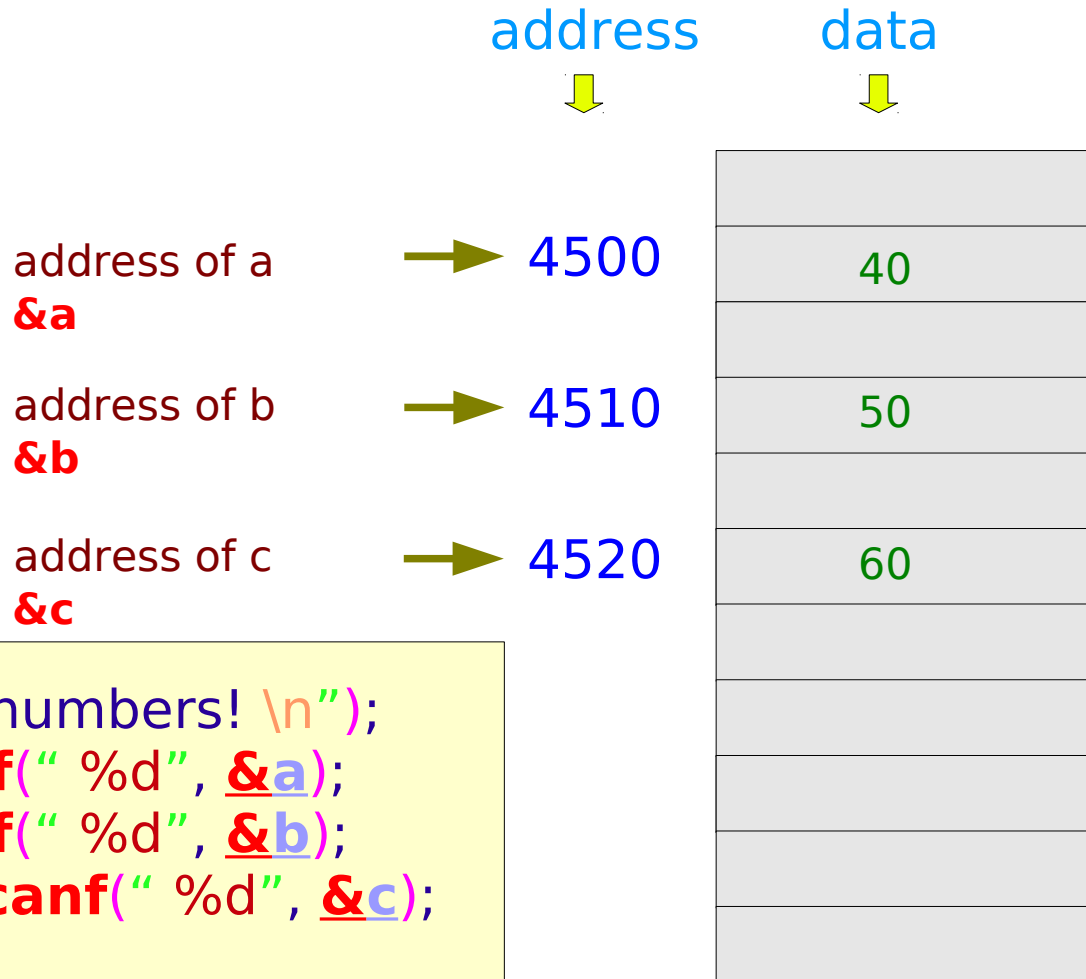
Expected Input

Enter three numbers!

a = 40 ↵

b = 50 ↵

c = 60 ↵



```
printf("Enter three numbers! \n");  
printf("a = "); scanf(" %d", &a);  
printf("b = "); scanf(" %d", &b);  
printf("c = ");   scanf(" %d", &c);
```

Main Function (2)

```
main (void)
```

```
{
```

```
    int    a, b, c;
```

```
    int    mean;
```

```
    printf("Enter three numbers! \n");
```

```
    printf("a = "); scanf(" %d", &a);
```

```
    printf("b = "); scanf(" %d", &b);
```

```
    printf("c = ");    scanf(" %d", &c);
```

```
    mean = (a + b + c) / 3;
```

```
    printf("The mean of three numbers \n");
```

```
    printf("a = %d \n b = %d \n c = %d \n", a, b, c);
```

```
    printf("mean (%d, %d, %d) => %d \n",  
           a,    b,    c,    mean );
```

```
}
```

Main Function (2)

```
main (void)
{
    int    a, b, c;
    int    mean;
```

```
    enter_numbers( ? );
```

```
    compute_mean( ? );
```

```
    print_numbers( ? );
```

```
}
```

Compute_mean Function

```
float compute_mean (int x, int y, int z)
{
    int    avg;

    avg = (x + y + z) / 3.0;

    return( avg );
}
```

- * Call by Value
- * Return Value
- * Local Variable

```
main (void)
{
    int    mean;

    mean = compute_mean(40, 50, 60);
}
```


Enter_numbers Function

```
void enter_numbers (int *x, int *y, int *z)
{
    printf("Enter three numbers! \n");
    printf("a = "); scanf(" %d", x);
    printf("b = "); scanf(" %d", y);
    printf("c = ");    scanf(" %d", z);
}
```

```
main (void)
{
    int    a, b, c;

    enter_numbers(&a, &b, &c);
}
```

* Call by Reference
* No Return Value

Print_numbers Function

```
void print_numbers (int x, int y, int z, float avg)  
{  
    printf("The mean of three numbers \n");  
    printf("a = %d \n b = %d \n c = %d \n", x, y, z);  
    printf("mean (%d, %d, %d) => %d \n",  
           x, y, z, avg );  
}
```

```
main (void)  
{  
    int    a, b, c;  
    float  mean;  
  
    print_numbers(a, b, c, mean);  
}
```

* Call by Value
* Return Value

Main Function (3)

```
void enter_numbers (int *x, int *y, int *z);  
float compute_mean (int x, int y, int z);  
void print_numbers (int x, int y, int z, float avg);
```

```
main (void)  
{  
    int    a, b, c;  
    int    mean;  
  
    enter_numbers(&a, &b, &c);  
  
    mean = compute_mean(a, b, c);  
  
    print_numbers(a, b, c, mean);  
}
```

References

- [1] Essential C, Nick Parlante
- [2] Efficient C Programming, Mark A. Weiss
- [3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
- [4] C Language Express, I. K. Chun