Array (1A)

Young Won Lim
7/21/09

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

This document was produced by using OpenOffice.

Calculating the Mean of n Numbers

The mean of n numbers

$$m = \frac{\sum_{i=0}^{n-1} x_i}{n}$$

$$m = \frac{\sum_{i=0}^{4} x_i}{5} = \frac{(x_0+x_1+x_2+x_3+x_4)}{5}$$

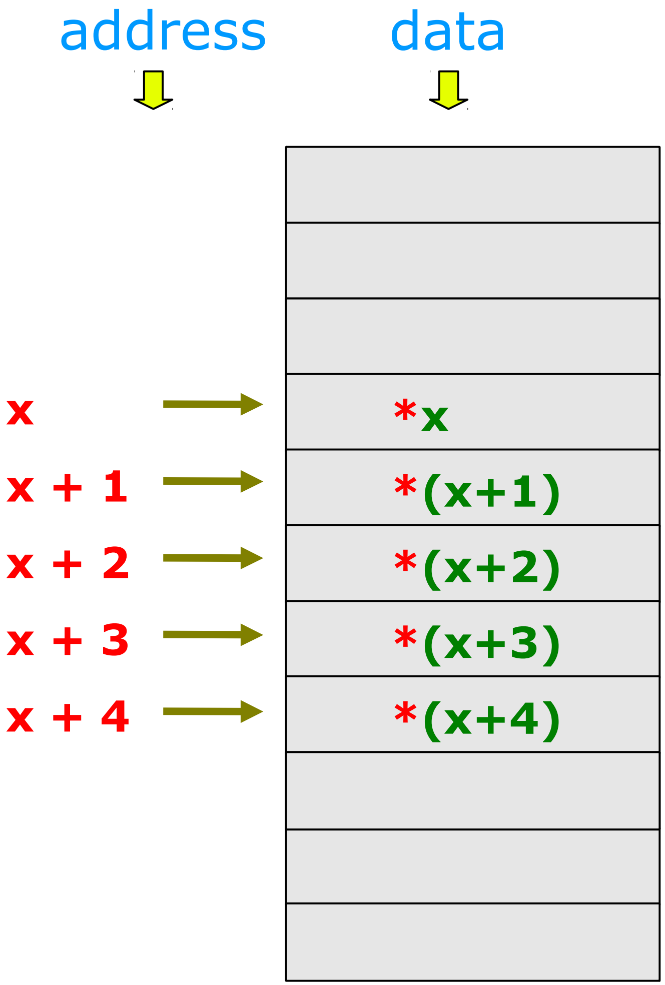
Array and Memory

```
int        x[10];
```

x holds *address*
*to **10** consecutive **int** variables*

10 int variables

index	data
0	x[0]
1	x[1]
2	x[2]
3	x[3]
4	x[4]



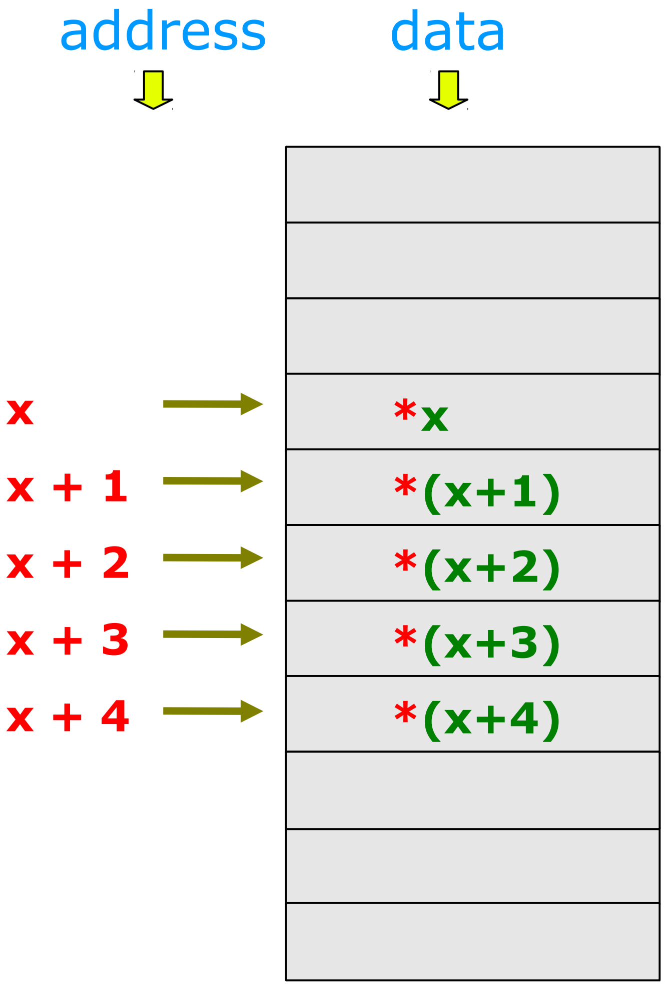
Array and Memory

```
int        x[10];
```

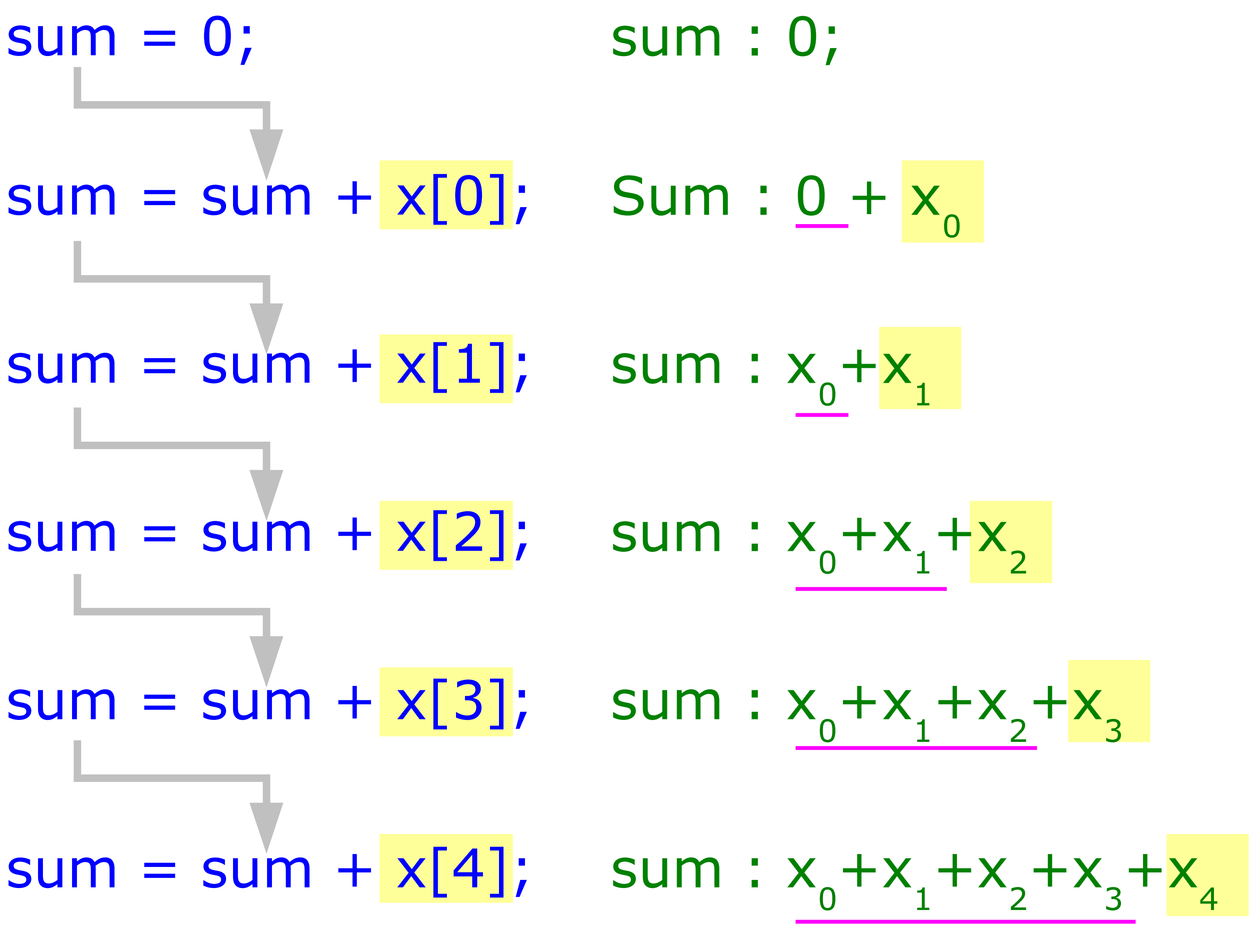
x holds *address*
*to **10** consecutive **int** variables*

10 int variables

x[0] = 80	80	*(x+0) = 80
x[1] = 90	90	*(x+1) = 90
x[2] = 40	40	*(x+2) = 40
x[3] = 70	70	*(x+3) = 70
x[4] = 60	60	*(x+4) = 60



Computing the sum of n numbers (1)



```
sum = 0;

sum = sum + x[0];

sum = sum + x[1];

sum = sum + x[2];

sum = sum + x[3];

sum = sum + x[4];
```

sum : 0;

Sum : $\underline{0} + x_0$

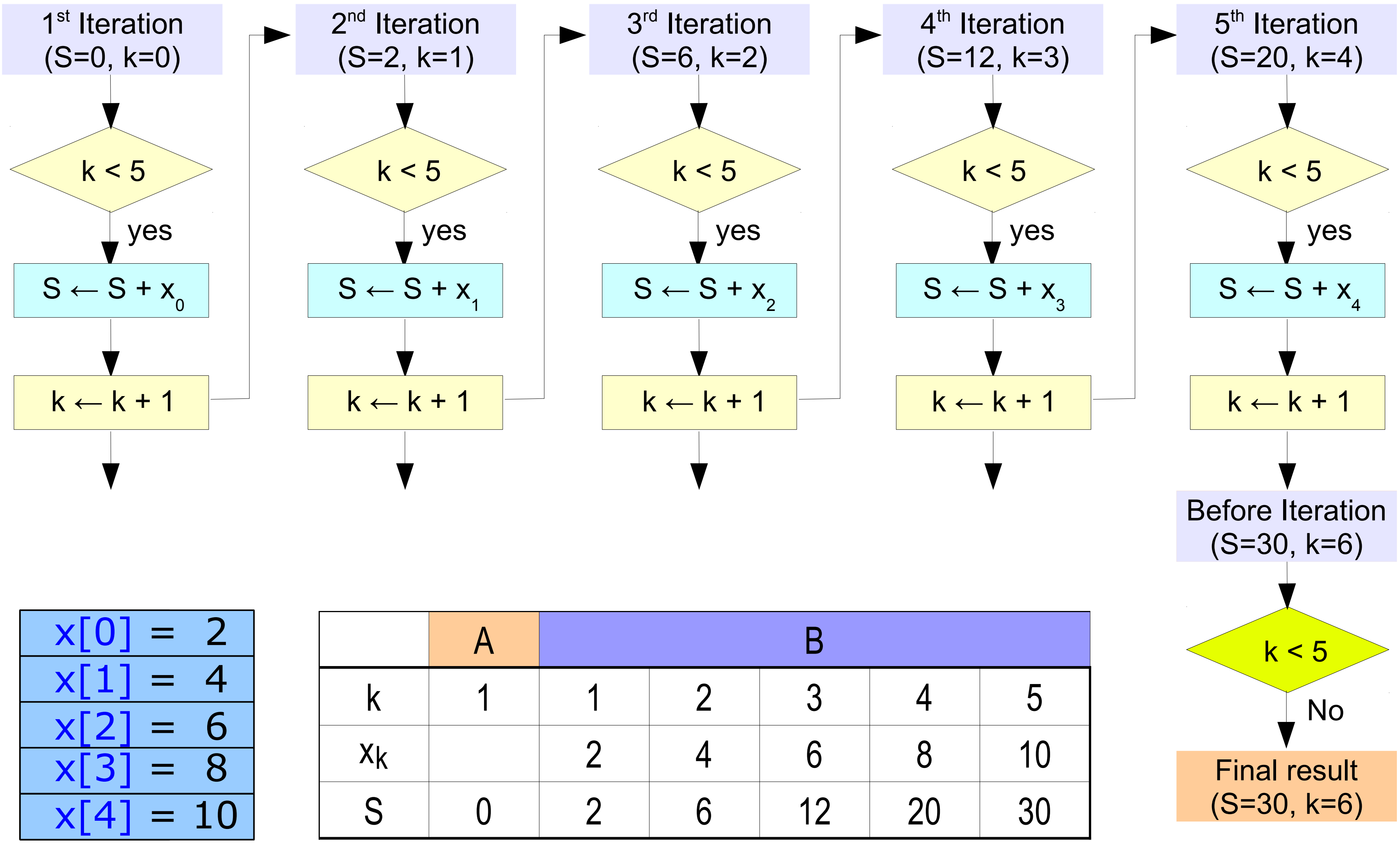
sum : $\underline{x_0} + x_1$

sum : $\underline{x_0 + x_1} + x_2$

sum : $\underline{x_0 + x_1 + x_2} + x_3$

sum : $\underline{x_0 + x_1 + x_2 + x_3} + x_4$

Computing the sum of n numbers (2)



x[0]	=	2
x[1]	=	4
x[2]	=	6
x[3]	=	8
x[4]	=	10

	A	B				
k	1	1	2	3	4	5
x_k		2	4	6	8	10
S	0	2	6	12	20	30

Computing the sum of n numbers (3)

```
int main (void)
{
        int x[5] = {2, 4, 6, 8, 10};
        int k, sum = 0;

        for (k=0; k<5; ++k)
                sum += x[k];

        printf("sum = %d \n", sum);

        return 0;
}
```

int x[5];

x holds *address*
*to **5** consecutive **int** variables*

x[🟦]

x is the name of an array
x holds the address
of the first element of that array
🟦 *is the number of array elements*

int *is the type of each array element*

Computing the sum of n numbers (4)

```
int arr_sum ( int a[] );

int main (void)
{
        int x[5] = {2, 4, 6, 8, 10};
        int k, sum = 0;

        sum = arr_sum ( x[] );

        printf("sum = %d \n", sum);

        return 0;
}

int arr_sum ( int a[] )
{
        int k, sum = 0;

        for (k=0; k<5; ++k)
                sum += a[k];

        return sum;
}
```

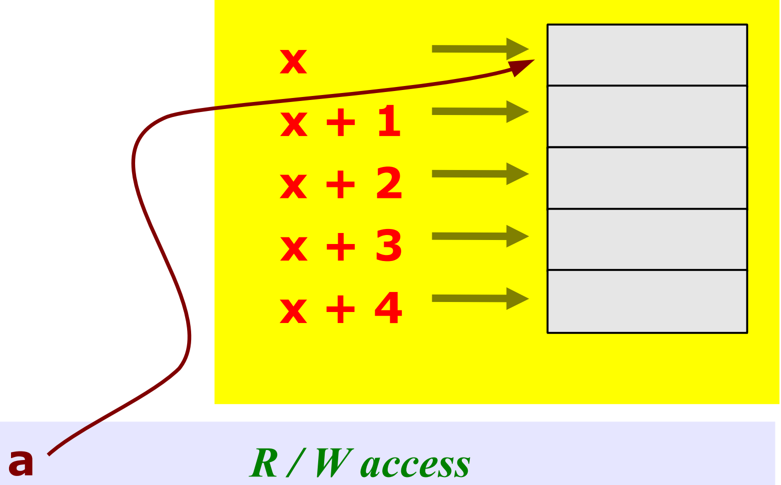
function call in main

sum = **arr_sum** (x[ ]);

⬆ *out* ⬇ *in*

int **arr_sum** (int a[ ]);

function declaration



R / W access

In the function, the array defined in main() can be accessed through the array name

call by reference

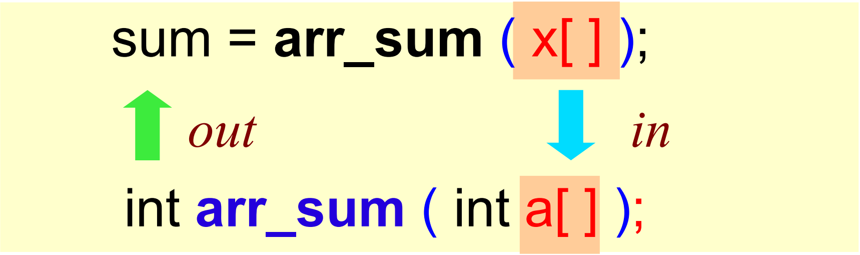
Computing the sum of n numbers (5)

```
int arr_sum ( int a[] );
```

```
int main (void)
{
        int x[5] = {2, 4, 6, 8, 10};
        int k, sum = 0;

        sum = arr_sum ( x[] );

        printf("sum = %d \n", sum);

        return 0;
}
```

```
int arr_sum ( int * a )
{
        int k, sum = 0;

        for (k=0; k<5; ++k)
                sum += a[k];

        return sum;
}
```

function call in main



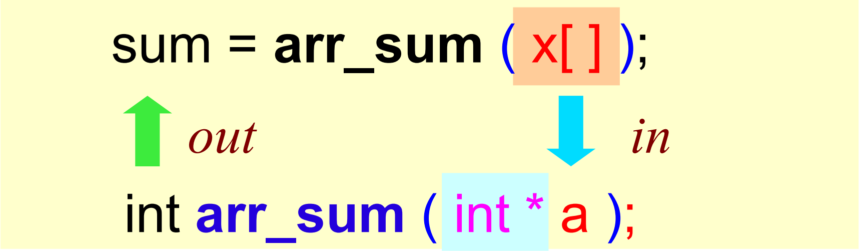
```
sum = arr_sum ( x[ ] );
int arr_sum ( int a[ ] );
```

function declaration

Array name is passed as an input

Array name contains the address of the 1st element of an array

Function can receive this array name as a pointer to integer



```
sum = arr_sum ( x[ ] );
int arr_sum ( int * a );
```

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