One Dimensional Array in C

Array

- List of variables of same type
- Accessed through a common name
- General form
  - type var_name[size];
  - int myarray[20];
  - int m[10], a[5];
- Accessed by indexing
  - Known as subscript
  - Can be any valid expression
  - Begin at 0
  - myarray[1] : 2nd element
- Array elements are stored in contiguous memory location

Array (Initialization)

- int n[6]={48, 53, 26,71, 9, 12};

48	**53**	**26**	**71**	**9**	**12**

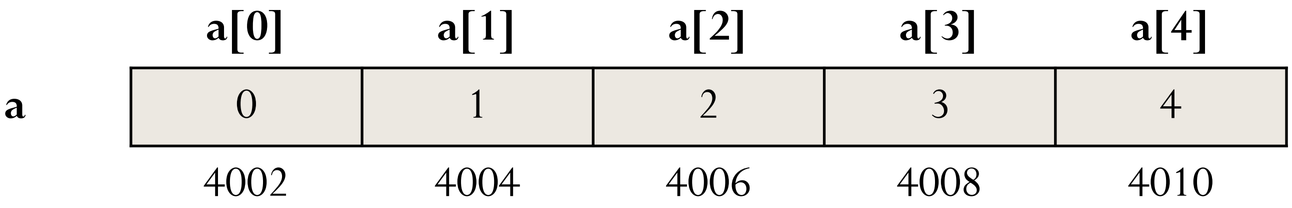
- float n[]={3.1, -5, 2.5, 17.4, 29};
  - **Array dimension optional**

3.1	**-5**	**2.5**	**17.4**	**29**

Array

```
int a[5];
for(int i=0; i<5; i++) {
        a[i]=i;
}
```

- After the declaration
  - 10 bytes get reserved in memory
  - Each integer 2 bytes long



	a[0]	**a[1]**	**a[2]**	**a[3]**	**a[4]**
a	0	1	2	3	4
	4002	4004	4006	4008	4010

Array

- Can be used anywhere a variable/constant can

```
int a[5];
for(int i=0; i<5; i++) {
            scanf("%d", &a[i]);
}
```

Array

- C does not perform any bound checking on array index
- Program may crash

```
int a[5];
for(int i=0; i<5; i++) {
        a[i]=i;
}
printf("%d\n", a[10]);
```

Array

- It is not possible to assign one entire array to other array

```
int a1[5], a2[5];
a1=a2; //not possible
```

- Need to copy explicitly

```
for(int i=0; i<5; i++) {
        a1[i]=a2[i];
}
```

Array (Average Calculation)

```
#include<stdio.h>
int main(){
        int a[5];
        double sum=0;
        for(int i=0; i<5; i++) {
                scanf("%d", &a[i]);
        }
        for(i=0; i<5; i++) {
                sum=sum+a[i];
        }
        printf("Average is %lf\n", sum/5.0);
        return 0;
}
```