Lecture 14

Array: Array is a list of variables, which are all of the same type and are referenced through a common name. An individual variable in the array is called an array element.

Advantages of array:

- Reduce ordinary variable
- Allocates memory sequentially

Disadvantages of array:

- Insertion difficult
- Deletion difficult
- Wastage of memory

Array declaration: We can declare an array in two ways.

- Static array
- Dynamic array

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

x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]	x[9]

This is called static declaration of an array.

```
int *p, n;
scanf("%d",&n);
p = (int * ) malloc (n * sizeof(int));
```

This is called dynamic declaration of an array. By using (void*)malloc(int size) we can declare dynamic array.

One dimensional arrays: Say, int a[5];

Example 1:

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

```
1
2
3
4
5
12345
```

Example 2:

```
void main()
{
        char *p, n, i ;
        scanf("%d",&n);
        p = (char *) malloc (n);
        for ( i = 0 ; i < n ; i ++)
                scanf("%d", ( p + i ));
        for ( i = 0 ; i < n ; i ++)
                printf("%d", *( p + i ));
}
```

```
5
1
2
3
4
5
12345
```

Example 3:

```
void main()
{
        int days[31], n, avg = 0, i ;
        printf(" How many days ? \n");
        scanf("%d",&n);
        for ( i = 0 ; i < n ; i ++){
                scanf("%d", & day [i] );
                avg = avg + day [i] ;
        }
        printf(" Average temperature = %f", (float)avg/n);
}
```

Consider the declaration, int (*any)(int, int).
This declaration indicates that any is a pointer of function which can take two integer type values as argument and also return an integer type value.

Example 4:

```
int sum(int a, int b);
void main()
{
        int a, b;
        int (*sab)(int, int);
        sab = sum;
        scanf("%d%d", &a, &b);
        printf("\n%d",sab(a,b));
}

int sum (int a, int b)
{
        return a+b;
}
```

```
5
4
9
```

Example 5:

```
void main()
{
        int a[6], sqr_a[6], *p, *q ;
        p = a ;
        q = sqr_a;
        for( i = 0 ; i < 5 ; i ++){
                *p = i + 1 ;
                p ++ ;
                *q = (i + 1)*(i + 1);
                q ++ ;
        }
        for( i = 0 ; i < 5 ; i ++)
                printf("%d/t%d\n"a[i],sqr_a[i]);
}
```

```
1    1
2    4
3    9
4    16
5    25
```

Passing array to a function: We can pass the whole array to a function by just sending the first element of that array.

Example 6:

```
void input(int *);
void output(int []);

void main()
{
        int  a[10];
        input  (a);
        output(a);
}

void input(int *a)
{
        int i;
        for( i = 0 ; i < 10 ; i ++){
                scanf("%d", (a+i));
}

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

```
1
2
3
4
5
6
7
8
9
10
12345678910
```