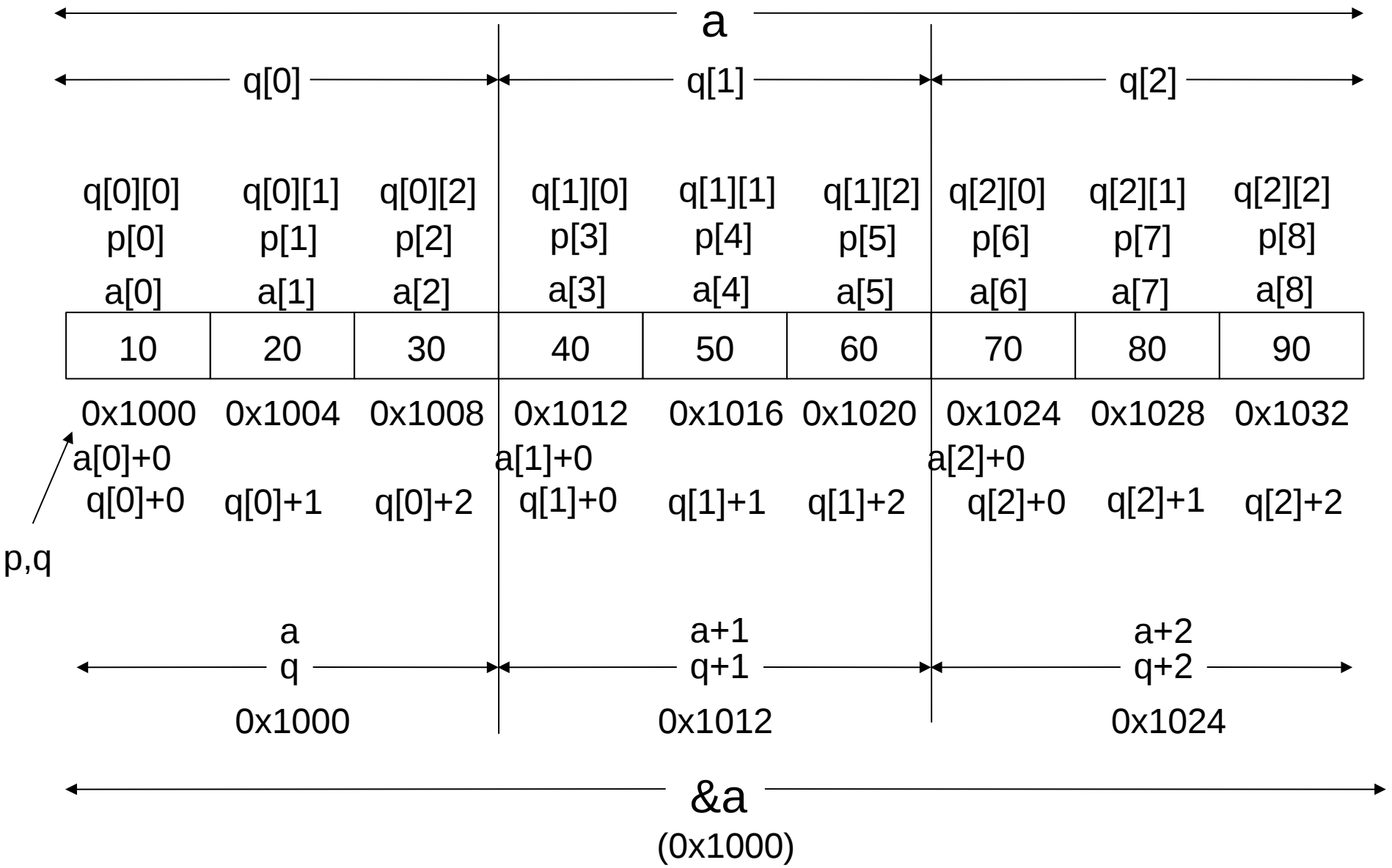
Pointer to Array

A pointer , which is pointing to a complete array is called pointer to array. That means a pointer increments/decrements based on complete array size.

Syntax :

Datatype (*ptr)[size] = Array_address;

Pointer to an array is implemented to point to 2D-array Base address and access the elements.


a
q[0]
q[1]
q[2]
q[0][0]
p[0]
a[0]
q[0][1]
p[1]
a[1]
q[0][2]
p[2]
a[2]
q[1][0]
p[3]
a[3]
q[1][1]
p[4]
a[4]
q[1][2]
p[5]
a[5]
q[2][0]
p[6]
a[6]
q[2][1]
p[7]
a[7]
q[2][2]
p[8]
a[8]
10
20
30
40
50
60
70
80
90
0x1000
0x1004
0x1008
0x1012
0x1016
0x1020
0x1024
0x1028
0x1032
a[0]+0
q[0]+0
q[0]+1
q[0]+2
a[1]+0
q[1]+0
q[1]+1
q[1]+2
a[2]+0
q[2]+0
q[2]+1
q[2]+2
p,q
a
q
0x1000
a+1
q+1
0x1012
a+2
q+2
0x1024
&a
(0x1000)

```
1 #include<stdio.h>
2 int main()
3 {
4         int a[] = {10,20,30,40,50,60,70,80,90};
5         int *p;
6         int (*q)[3];
7         p = a; //pointer to an integer
8         q = (int(*)[3])&a; //pointer to an array
9         printf("%d  %d\n",sizeof p,sizeof q);
10
11         printf("p = %p  p+1 = %p\n",p,p+1); //1000 , 1004
12         printf("q = %p  q+1 = %p\n",q,q+1); //1000 , 1012
13
14 }
15         //p = a; --> int * = int *
16         //q = a; --> int (*)[3] = int * //warning
17         //&a --> int (*)[9];
18         //q = &a; --> int (*)[3] = int (*)[9] //warning
19
```

```
 1 #include<stdio.h>
 2 int main()
 3 {
 4         int a[9] = {10,20,30,40,50,60,70,80,90};
 5         int *p;
 6         int (*q)[3];
 7
 8         p = a;
 9         q = (int (*)[3])a;
10          printf("*p = %d\n",*p);  //10
11          printf("*q = %p\n",*q);  //0x1000
12          printf("**q = %d\n",**q);  //10
13 }
```

```
 1 #include<stdio.h>
 2 int main()
 3 {
 4         int a[10] = {10,20,30,40,50,60,70,80,90,100};
 5         int *p;
 6         int (*q)[3];
 7
 8         p = a;
 9         q = (int (*)[3])a;
10          printf("q = %u\n",(unsigned)q); //0x1000
11          printf("q+1 = %u\n",(unsigned)(q+1));//0x1012
12          printf("*q = %u\n",(unsigned)*q);//0x1000
13          printf("**q = %d\n",**q); // 10
14          printf("*q[0] = %d\n",*q[0]);// 10
15          printf("(*q)[0] = %d\n",(*q)[0]);//10
16 }
//    *q[0] ---> (*q)[0] --> q[0][0]
```

```
1 #include<stdio.h>
2 int main()
3 {
4         int a[3][3] = { {10,20,30}, {40,50,60}, {70,80,90} };
5         printf("a = %u\n",(unsigned)a);
6         int (*q)[3];
7         q = a;  //int (*)[3] = int (*)[3];
8
9         printf("q = %p q+1 = %p\n",q,q+1);//0x1000, 0x1012
10         printf("*q[0] = %d\n",*q[0]); //10
11         printf("(*q)[0] = %d\n",(*q)[0]);//10
12         printf("*q[1] = %d\n",*q[1]);//q[1][0]-->40
13         printf("(*q)[1] = %d\n",(*q)[1]);//q[0][1]-->20
14 }
15 //*q[1] ---> *(q[1]+0) ---> q[1][0];
16 //(*q)[1]--> q[0][1] --> 20;
```

```
1 //Write a program to access 2D array elements using pointer to an array.
2 #include<stdio.h>
3 int main()
4 {
5    int a[3][3] = { {10,20,30}, {40,50,60}, {70,80,90} };
6    int (*q)[3] = a;
7    int i,j;
8 /*
9    for(i=0;i<3;i++)
10   {
11        for(j=0;j<3;j++)
12        printf("%d ",q[i][j]);
13        printf("\n");
14   }
15 */
16   for(i=0;i<3;i++)
17   {
18        for(j=0;j<3;j++)
19        printf("%d ",(*q)[j]);  //q[0][j]
20        printf("\n");
21
22        q++;
23   }
```

```
1 //write a program to pass 1D array to a function.
2 #include<stdio.h>
3 void fun(int *p);
4 int main()
5 {
6         int a[] = {10,20,30,40,50};
7         fun(a);
8 }
9 void fun(int *p)
10 {
11          int i;
12          for(i=0;i<5;i++)
13          //printf("%d  ",p[i]);
14          printf("%d  ",*p++);
15
16          printf("\n");
17 }
```

```
1 //write a program to pass 2D array to a function.
2 #include<stdio.h>
3 void fun(int r,int c,int (*p)[c]);
4 int main()
5 {
6         int r,c;
7         int a[2][3] = {{10,20,30},{11,22,33}};
8         r = sizeof a/sizeof a[0];
9         c = sizeof a[0]/sizeof a[0][0];
10         fun(r,c,a);
11 }
12 void fun(int r,int c,int (*p)[c])
13 {
14         int i,j;
15         for(i=0;i<r;i++)
16         {
17                 for(j=0;j<c;j++)
18                 printf("%d ",(*p)[j]);
19                 printf("\n");
20                 p++;
21         }
22 }
```

2nd method :

```
void fun(int r,int c,int p[r][c])  //p[r][c] --> (*p)[c]
{
        int i,j;
        for(i=0;i<r;i++)
        {
                for(j=0;j<c;j++)
                printf("%d ",p[i][j]);
                printf("\n");
        }
}
```

```
1 //write a program to pass 2D array to a function. (3rd method)
2 #include<stdio.h>
3 void fun(int r,int c,int *p);
4 int main()
5 {
6         int r,c;
7         int a[2][3] = {{10,20,30},{11,22,33}};
8         r = sizeof a/sizeof a[0];
9         c = sizeof a[0]/sizeof a[0][0];
10         //fun(r,c,(int *)a);
11         fun(r,c,*a);
12 }
13 void fun(int r,int c,int *p)
14 {
15         int i,j;
16         for(i=0;i<r;i++)
17         {
18                 for(j=0;j<c;j++)
19                 printf("%d  ",*p++);
20                 printf("\n");
21         }
22 }
```