Array of pointer :

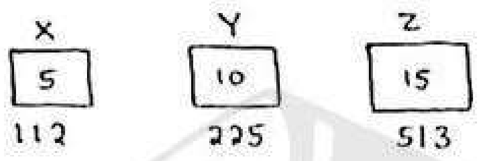
→ " The array of pointer is an array that stores the pointers."

→ Syntax : $\boxed{\text{data-type * pointer-name [SIZE] ;}}$

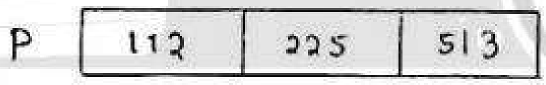
→ For example, int * p[3] ;

→ Example : let there are 3 variables.

X = 5   Y = 10   Z = 15



if we declare int * p[3], then p is an array of pointer, which holds the addresses of variable x, y, z in an array. so



Here
```
P[0] = &X ;
P[1] = &Y ;
P[2] = &Z ;
```

→ An array of pointers will hold the collection of address

→ The address stored in the array of pointers can be addresses of isolated variables or addresses of array elements or any other addresses

→ All rules apply to an ordinary array also apply to the array of pointer

→ The array of pointer can be declared by preceding asterisk to an array.

```
/* Array of pointer */
#include.<stdio.h>
#include <conio.h>
 void main( )
  {
     int arr[5] = {10, 15. 20. 25. 30};
   int  *ptr[5] = {arr, arr+1, arr+2, arr+3, arr+4};
   int i;
   clrscr();
  for ( i=0; i<5; i++)
       printf ( Addr = %u  Addr = %u  value = %d ",
                           *(ptr+i), &arr[i], arr[i]);
   getch();
 }
```

output :
```
Addr = 4132   Addr = 4132   value = 10
Addr = 4134   Addr = 4134   value = 15
Addr = 4136   Addr = 4136   value = 20
Addr = 4138   Addr = 4138   value = 25
Addr = 4140   Addr = 4140   value = 30
```

Pointer to two-dimensional array :

→ If in a two dimensional array (matrix) . C = no. of column , then

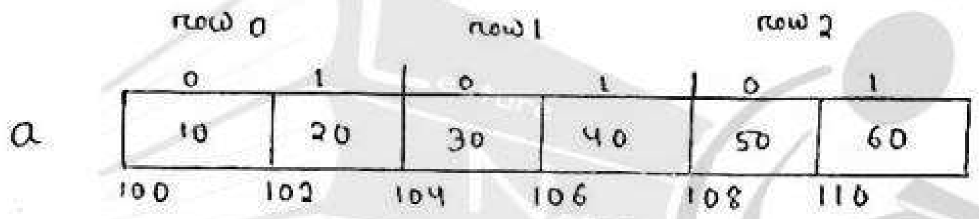
$$\boxed{\text{Address of } a[i][j] = a + (i * C + j) \cdot \text{sizeof(datatype)}}$$

→ Example . let a[3][2] = {

row = 3

column = 2

```
                    {10, 20},
                    {30, 40},
                    {50, 60}
                  };
```



Here the address of a[2][1] =

$$100 + (2 * 2 + 1) \times 2$$
$$= 100 + 10$$
$$= 110$$

```
/* Accessing two-dimensional array elements using pointer */
#include <stdio.h>
#include <conio.h>
void main()
{
   int a[10][10], i, j, r, c, *P;
    clrscr();
   printf("Enter the order of matrix a :");
scanf ("%d %d", &r, &c);
printf (" Enter %d value to the matrix a", r*c);
for( i=0; i<r; i++)
  for ( j=0; j<c; j++)
      scanf ("%d", &a[i][j]);
p = a ; or  p = &a[0][0];
printf (" The resultant matrix a is :\n")
    for (i=0; i<r; i++)
      { for ( j=0; j<c; j++)
          {
           printf ("%d", *(p+i*c+j));
          }
        printf ("\n");
      }
 getch();
}
```

output :
```
Enter the order of matrix a : 2 2
Enter 4 value to the matrix a : 10 05 04 15

The resultant matrix a is :

        10    05
        04    15
```

```
/* program to convert a two-dimensional array into a
   single dimensional array */
# include <stdio.h>
# include <conio.h>
void main()
{
    int a[3][3], b[9], i, j, *p;
   clrscr();
   printf("Enter 9 numbers:");
       for( i=0; i<3; i++)
         for (j=0; j<3; j++)
              scanf("%d", &a[i][j]);
  p = &a[0][0];
  for( i=0; i<9; i++)
       b[i] = *p++;
  printf(" Result in one dimensional array is :\n");
    for(i=0; i<9; i++)
         printf(" %d", b[i]);
  getch();
}
```

output : Enter 9 elements : 2 3 4 5 6 7 8 9 10

```
Result in one dimensional array is :
   2 3 4 5 6 7 8 9 10
```

```
/* program to add two 3*3 matrices using pointer */

#include <stdio.h>
# include <conio.h>
void main()
{
   int a[3][3], b[3][3], c[3][3], i, j, *p, *q, *r;
   clrscr();
   prinf (" Enter the elements of the matrix a:");
    for (i=0; i<3 ; i++)
    for ( j=0; j<3 ; j++)
         scanf (" %d", &a[i][j]);
 prinlf (" Enter the elements of the matrix b:");
       for (i=0 ; i<3 ; i++)
       for (j=0; j<3 ; j++)
           scanf (" %d", &b[i][j]);
prinlf (" Matrix a is :");
          for (i=0; i<3; i++)
           { for (j=0; j<3; j++)
               { prinlf (" %d" , a[i][j]);
               }
             prinlf ("\n");
           }
prinlf (" Matrix b is :");
            for (i=0; i<3; i++)
                { for (j=0; j<3 ; j++)
                      { prinlf (" %d", b[i][j]);
                      }
                  prinlf ("\n");
                }
 p = &a[0][0];
 q = &b[0][0];
for (i=0; i<3; i++)
     for (j=0; j<3; j++)
        c[i][i] = 0;
    r = &c[0][0];
for (i=0; i<3; i++)
   for (j=0; j<3; j++)
      {
         *r = *p + *q;
          p++;
          q++;
          r++;
      }
 printf(" The resultant matrix c is :\n");
      r = r-9;
  for (i=0; i<3; i++)
      {  for (j=0; j<3; j++)
          {  printf(" %d", *r);
              r++ ;
          }
          printf("\n");
      }
   getch();
}
```