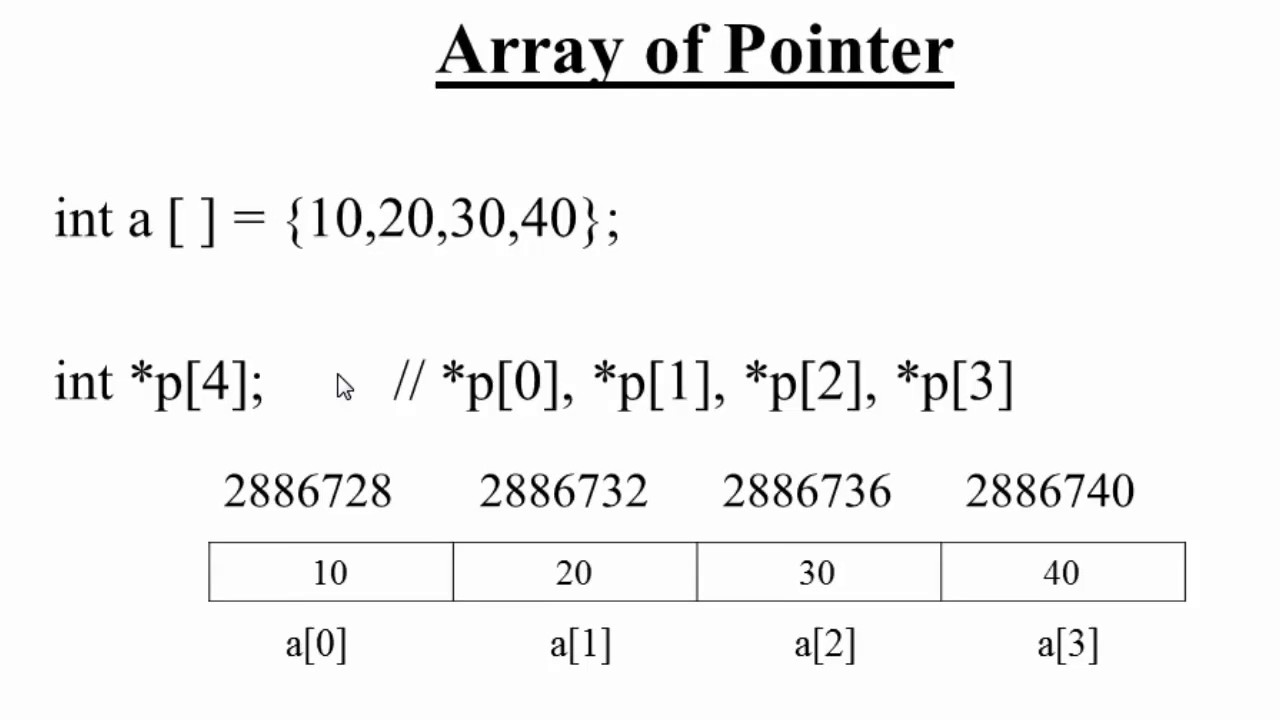
**Array of pointers**



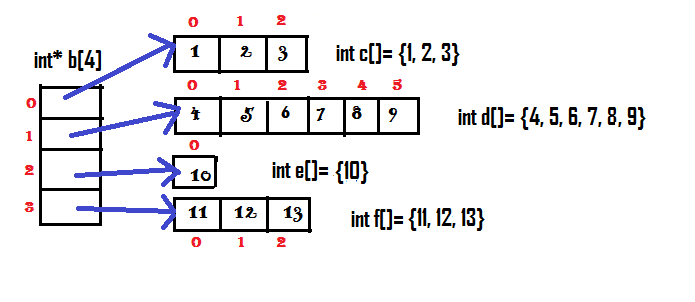
**P[i]=&a[i];**

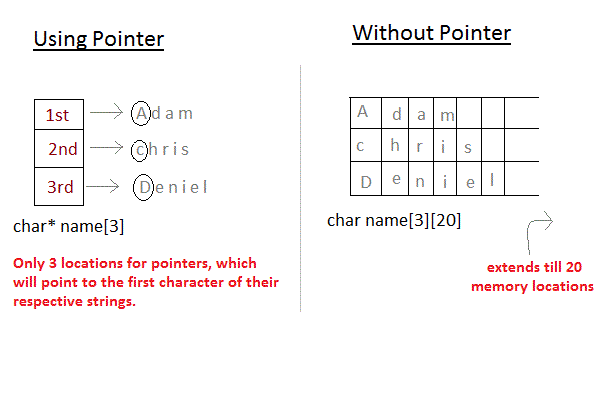
**P[i] will give the address of a[i];**

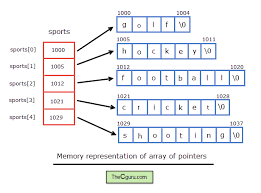
**\*P[i] will give the value at address of a[i]**

**&a[0] --------P[0]**

**a[0] --------\*(P[0])**







Where  **ptr** as an array of MAX integer pointers. Thus, each element in ptr, holds a pointer to an int value. The following example uses three integers, which are stored in an array of pointers, as follows −

[Live Demo](http://tpcg.io/UJoygb)

#include <stdio.h>

const int MAX = 3;

int main () {

int var[] = {10, 100, 200};

int i, \*ptr[MAX];

for ( i = 0; i < MAX; i++) {

ptr[i] = &var[i]; /\* assign the address of integer. \*/

}

for ( i = 0; i < MAX; i++) {

printf("Value of var[%d] = %d\n", i, \*ptr[i] );

}

return 0;

}

When the above code is compiled and executed, it produces the following result −

Value of var[0] = 10

Value of var[1] = 100

Value of var[2] = 200

You can also use an array of pointers to character to store a list of strings as follows −

[Live Demo](http://tpcg.io/UJBnBq)

#include <stdio.h>

const int MAX = 4;

int main () {

char \*names[] = {

"Zara Ali",

"Hina Ali",

"Nuha Ali",

"Sara Ali"

};

int i = 0;

for ( i = 0; i < MAX; i++) {

printf("Value of names[%d] = %s\n", i, names[i] );

}

return 0;

}

When the above code is compiled and executed, it produces the following result −

Value of names[0] = Zara Ali

Value of names[1] = Hina Ali

Value of names[2] = Nuha Ali

Value of names[3] = Sara Ali

