int* p;
-> 'p' is pointer pointing to integer

```
int *ptr(int, int)
        or
int* ptr(int, int)
 3     1      2
```

First : Identity , pointer(*)
Second : ( ) , [ ]
Third : Datatype

"ptr" is an identity of function which is taking 2 integers and return address of integer variable.

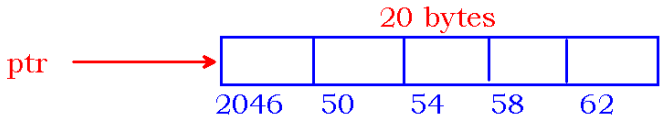
```
int*  add(int a, int b)
{
   int c = a+b;
   return  &c ;
}
```

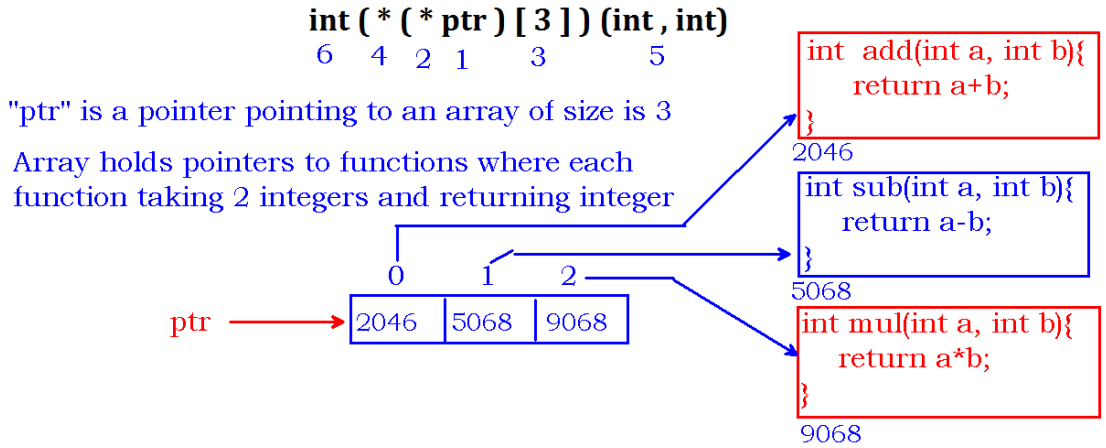
```
 4   2 1      3
int (*ptr)(int, int)
```

"ptr" is a pointer which is pointing to a function taking 2 integer arguments and returning int value

```
  4    2 1   3
float (*ptr) [5]
```

"ptr" is a pointer which is pointing to an array of size is 5 and holds float type data

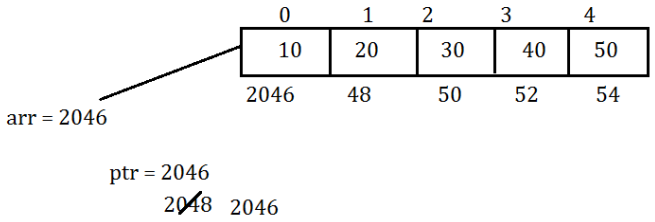




Accessing elements of array using expressions including pointers:

First Priority: ( )
Second Priority: ++ , --
Third Priority: Pointer(*)
Last priority: Arithmetic operators

```
#include<stdio.h>
int main()
{
    int arr[5] = {10,20,30,40,50},i;
    int* ptr;
    ptr = arr;
    printf("%u\n", *++ptr + 3);
    printf("%u\n", *(ptr-- + 2) + 5);
    printf("%u\n", *(ptr+3)-10);
    return 0;
}
```



First Priority: ( )
Second Priority: ++ , --
Third Priority: Pointer(*)
Last priority: Arithmetic operators

*++ptr + 3

* ++2046 + 3
* 2048 + 3
20 + 3
23

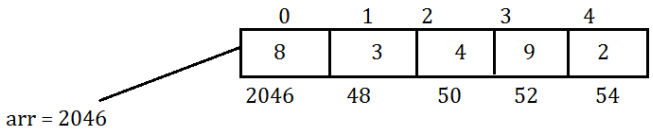
*(ptr-- + 2) + 5

*(2048-- + 2) + 5
*(2048 + 2) + 5
*(2052) + 5
40 + 5
45

*(ptr+3)-10

*(2046 + 3)-10
*(2052) - 10
40 - 10
30

```
#include<stdio.h>
void main()
{
	int arr[5] = {8, 3, 4, 9, 2},i;
	int* ptr;
	ptr = arr;
	printf("%u\n", *(--ptr+2) + 3);
	printf("%u\n", *(++ptr + 2) - 4);
	printf("%u\n", *(ptr-- +1 ) + 2);
}
```



ptr = 2046
2044
2046 2044

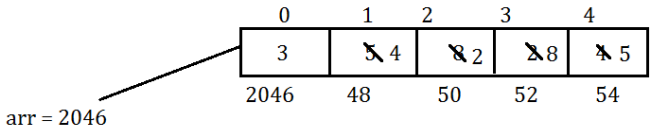
First Priority: ( )
Second Priority: ++ , --
Third Priority: Pointer(*)
Last priority: Arithmetic operators

*(--ptr + 2) +3
*(--2046 + 2) + 3
*(2044 + 2) + 3
*(2048) + 3
3 + 3
6

*(++ptr + 2) - 4
*(++2044 + 2) - 4
*(2046 + 2) - 4
*(2050) - 4
4 - 4
0

*(ptr-- + 1) + 2
*(2046-- + 1) + 2
*(2046 + 1) + 2
*(2048) + 2
3 + 2
5

```
#include<stdio.h>
void main()
{
	int arr[5] = {3,5,8,2,4},i,j,temp;
	int* ptr;
	ptr = arr;
	for(i=1, j=4 ; i<j ; i++ , j--)
	{
		temp = *(ptr+i);
		*(ptr+i) = *(ptr+j) ;
		*(ptr+j) = temp ;
	}
	printf("%d \n", *(++ptr+2)+2);
}
```



ptr = 2046

```
for(i=1 , j=4 ;    i<j  ;  i++ , j++)
{
	Swaping i, j location
		values
}
```

*(++ptr + 2) + 2
*(++2046 + 2) + 2
*(2048 + 2) + 2
*(2052) + 2
8 + 2
10

Note: Elements in function execution from left to right

```
#include<stdio.h>
void main()
{
	int arr[5] = {10,20,30,40,50},i;
	int* ptr;
	ptr = arr;
	printf("%u,%u,%u\n", *ptr-- + 1, *(ptr++ + 1) , *(ptr-- + 2)-5);
}
```

Array of pointers :

Array of pointers variable holds more than one element address.

```
#include<stdio.h>
```

```
void main()
{
	int iarr[5] = {10,20,30,40,50},i;
	int* ptr[5];

	for(i=0 ; i<5; i++)
	{
		ptr[i] = &iarr[i];
		//ptr[i] = iarr+i;
	}
	printf("array elements are \n");
	for(i=0 ; i<5; i++)
	{
		printf("%d\n", *ptr[i]);
		//printf("%d\n", *(*(ptr+i)));
	}
}
```

```
#include<stdio.h>
void main()
{
	int arr[5] = {10,20,30,40,50};
	int *ptr[5], i;
	for(i=0 ; i<5 ; i++)
	{
		*(ptr+i) = arr+i;
	}
	printf("%d\n", *(*(ptr++ + 2)+1)+3);
	printf("%d\n", *(*++ptr+1)+3);
}
```

```
#include<stdio.h>
void main()
{
	char* s = "Naresh" ;
	printf("%s \n", s);
	printf("%c \n", s);
	printf("%c \n", *s);
	printf("%c \n", *(s+3));
	printf("%c \n", *s+3);
}
```

```
#include<stdio.h>
void main()
{
	char* str = "learnown";
	printf("%c\n" , *(str++ + 2)+3);
```

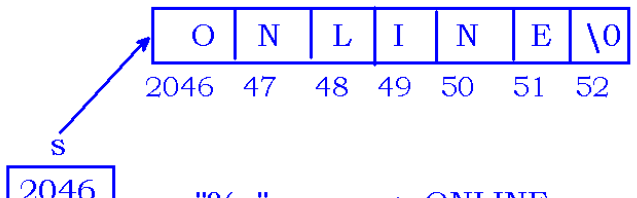
```
        printf("%c\n" , *++str+2);
        printf("%s\n" , --str-1);
}
```

```
#include<stdio.h>
void main()
{
        char* str = "learnown";
        printf("%c\n",*((str-- +2)+1)-3);
        printf("%c\n", *(--str + 3)-32);
        printf("%c\n",*(++str+2)+4);
}
```

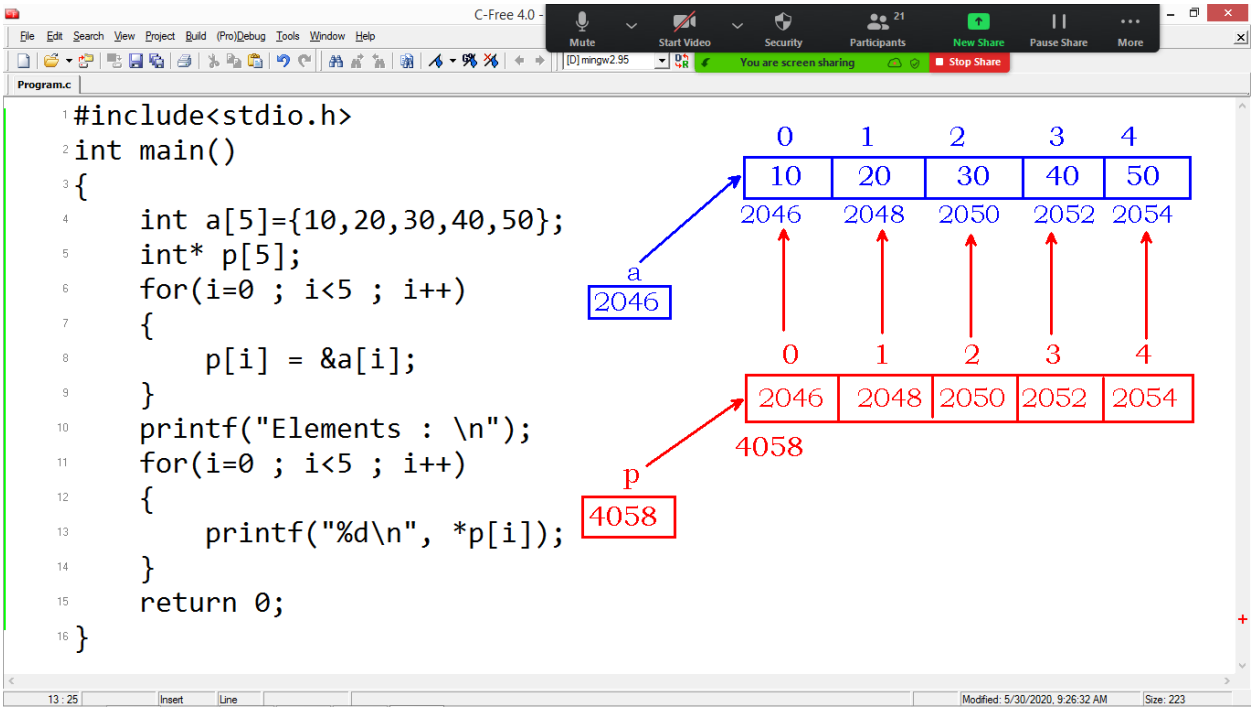
```
#include<stdio.h>
void main()
{
        char sport[ ]= "cricket";
        int x=1 , y;
        y=x++ + ++x;
        printf("%c",sport[++y]);
}
```

```
#include<stdio.h>
int main()
{
    char* s = "ONLINE";
    printf("%s \n", s);
    printf("%c \n", s);
    printf("%c \n", *s);
    printf("%c \n", *s+3);
    printf("%c \n", *(s+3));
    return 0;
}
```

%s -> Takes address to display
%c -> Takes location to display



"%s" , s ---> ONLINE
"%c" , s ---> garbage character
"%c" , *s ---> O
"%c" , *s+3 ---> M,N,O,P,Q,R
"%c" , *(s+3) --> *2049 -> I


C-Free 4.0
File Edit Search View Project Build (Pro)Debug Tools Window Help
Mute
Start Video
Security
Participants
New Share
Pause Share
More
You are screen sharing
Stop Share
Program.c
#include<stdio.h>
int main()
{
int a[5]={10,20,30,40,50};
int* p[5];
for(i=0 ; i<5 ; i++)
{
p[i] = &a[i];
}
printf("Elements : \n");
for(i=0 ; i<5 ; i++)
{
printf("%d\n", *p[i]);
}
return 0;
}
0 1 2 3 4
10 20 30 40 50
2046 2048 2050 2052 2054
a
2046
0 1 2 3 4
2046 2048 2050 2052 2054
4058
p
4058
13 : 25
Insert
Line
Modified: 5/30/2020, 9:26:32 AM
Size: 223