Calloc() : Allocate memory to array using pointers.

prototype :

void* calloc(size_t n , size_t size)

- First arg : Initial size of array.
- Second arg : Size of Array element.

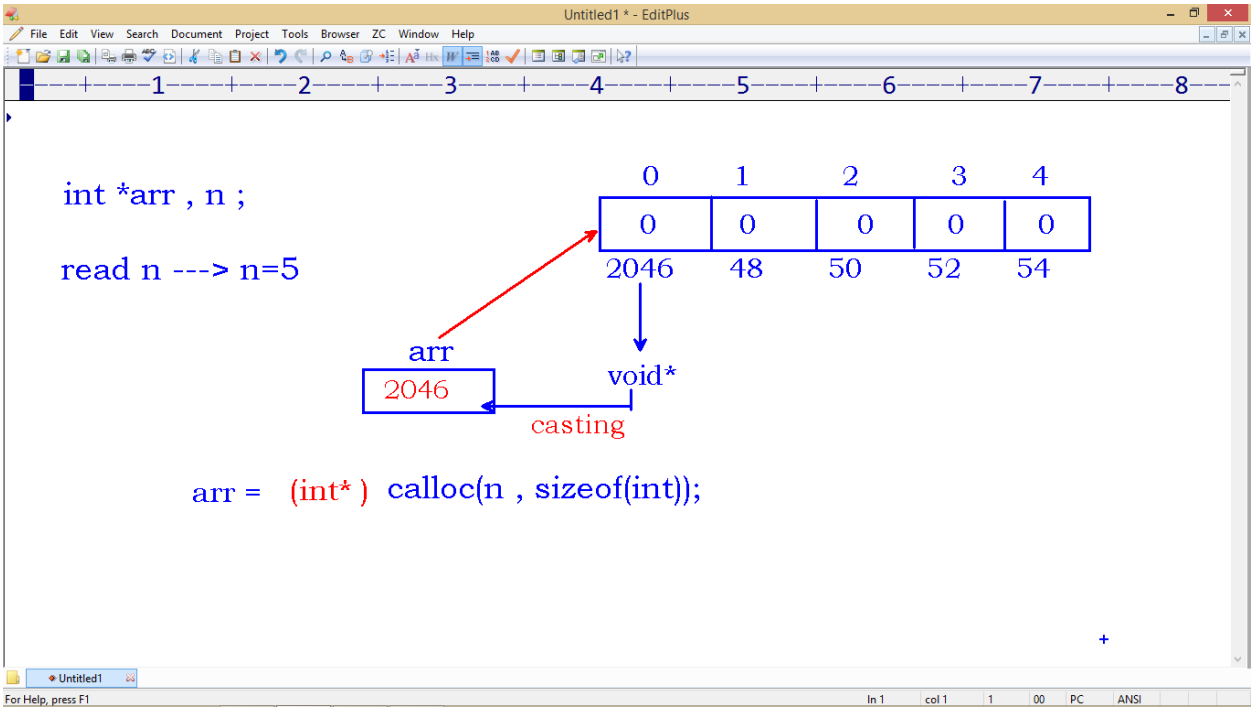
size_t : positive integer value

```
#include<stdio.h>
#include<string.h>
int main()
{
        size_t len ;
        len = strlen("Hello");
        printf("The length is : %u\n", len);
        return 0;
}
```

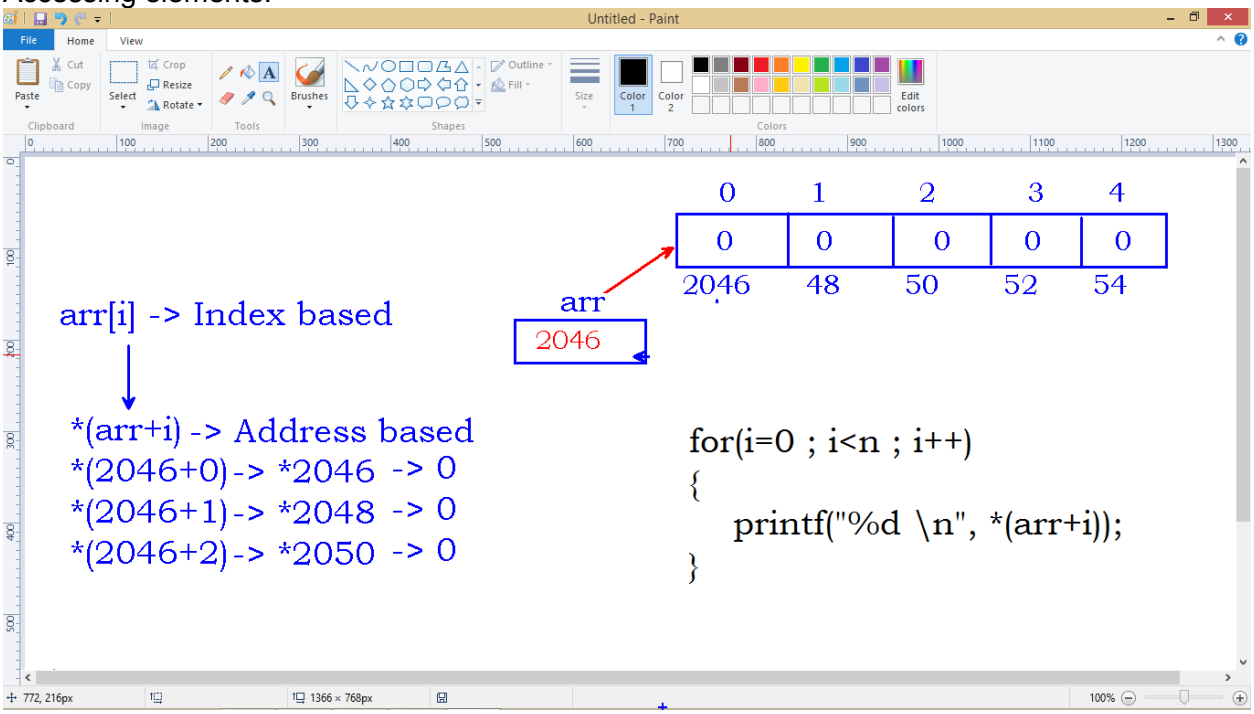
- On success - If memory is available, it allocates n*size bytes memory and returns base address of block.
- On failure - return NULL pointer

Why return type is void*?

-> void* is called Generic pointer.
-> calloc() function allocates memory to any type of pointer and it return common pointer void*.
-> We typecast to corresponding type.

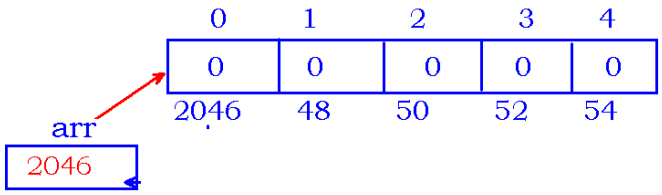


Accessing elements:



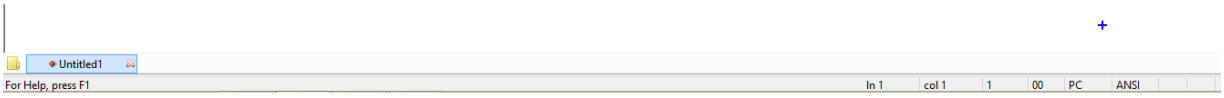

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
	int *arr, n;

	printf("Enter initial size of array : ");
	scanf("%d" , &n);

	arr = (int*)calloc(n , sizeof(int));
	if(arr==NULL)
	{
		printf("Out of memory \n");
	}
	else
	{
		int i;
		printf("Array elements are : \n");
		for(i=0 ; i<n ; i++)
		{
			printf("%d \n", *(arr+i));
		}
	}
	return 0;
}
```

Another way :



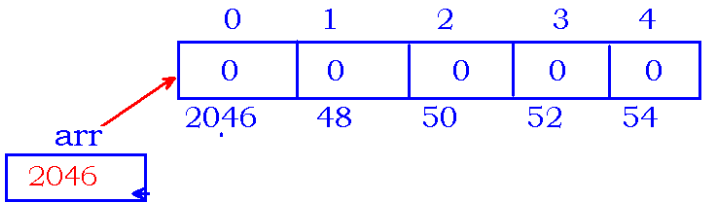
arr = 2046
n = 5
arr+n = 2056

```
int  *i;
for(i=arr ; i<arr+n ; i++)
{
    printf("%d \n", *i);
}
```




```
#include<stdio.h>
#include<stdlib.h>
int main()
{
	int *arr, n;

	printf("Enter initial size of array : ");
	scanf("%d" , &n);

	arr = (int*)calloc(n , sizeof(int));
	if(arr==NULL)
	{
		printf("Out of memory \n");
	}
	else
	{
		int *i;
		printf("Array elements are : \n");
		for(i=arr ; i<arr+n ; i++)
		{
			printf("%d \n", *i);
		}
	}
	return 0;
}
```

Reading elements:



scanf() function takes address to read the element into the block
(arr+i -> is an address)

```
for(i=0 ; i<n ; i++)
{
    scanf("%d", arr+i);
}
```

2046+0 = 2046
2046+1 = 2048
2046+2 = 2050

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
	int *arr, n;

	printf("Enter initial size of array : ");
	scanf("%d" , &n);

	arr = (int*)calloc(n , sizeof(int));
	if(arr==NULL)
	{
		printf("Out of memory \n");
	}
	else
	{
		int i;
		printf("Enter %d elements : \n", n);
		for(i=0 ; i<n ; i++)
		{
			scanf("%d", arr+i);
		}
		printf("Array elements are : \n");
		for(i=0 ; i<n ; i++)
		{
			printf("%d \n", *(arr+i));
		}
	}
	return 0;
}
```