# CS F111: Computer Programming

**(Second Semester 2021-22)**

**Lect 26: Pointers contd.**


Nikumani Choudhury

Asst. Professor, Dept. of Computer Sc. & Information System

BITS Pilani

Hyderabad Campus

# Output??

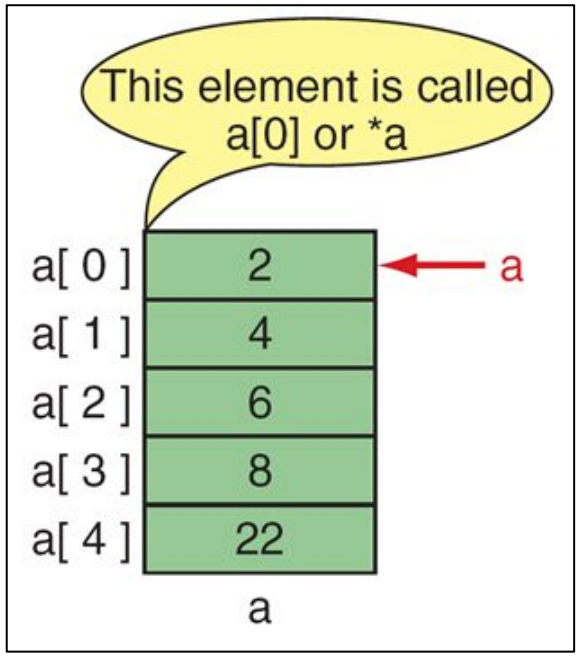
```c
#include <stdio.h>

int main()
{
    int *ptr;
    int x;

    ptr = &x;
    *ptr = 0;

    printf(" x = %dn", x);
    printf(" *ptr = %dn", *ptr);

    *ptr += 5;
    printf(" x  = %dn", x);
    printf(" *ptr = %dn", *ptr);

    (*ptr)++;
    printf(" x = %dn", x);
    printf(" *ptr = %dn", *ptr);

    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    float arr[5] = {12.5, 10.0, 13.5, 90.5, 0.5};
    float *ptr1 = &arr[0];
    float *ptr2 = ptr1 + 3;

    printf("%f ", *ptr2);
    printf("%d", ptr2 - ptr1);
   return 0;
}
```

```c
#include<stdio.h>
int main()
{
 char *ptr = "BITS Pilani";
 printf("%c", *&*&*ptr);
 return 0;
}
```

```c
#include<stdio.h>
int main()
{
    int arr[] = {10, 20, 30, 40, 50, 60};
    int *ptr1 = arr;
    int *ptr2 = arr + 5;
    printf("Number of elements between two pointer are: %d.\n",(ptr2 - ptr1));
    printf("Number of bytes between two pointers are: %d",(char*)ptr2 - (char*) ptr1);
    return 0;
}
```

# Dereference of Array name




```
#include <stdio.h>
int main() {
  int a[5] = {2, 4, 6, 8, 22};
  printf("%d %d %d", a[0], a[2], a[4]);
  printf("\n%d %d %d", *(a+0), *(a+2), *(a+4));
return 0;
}
```

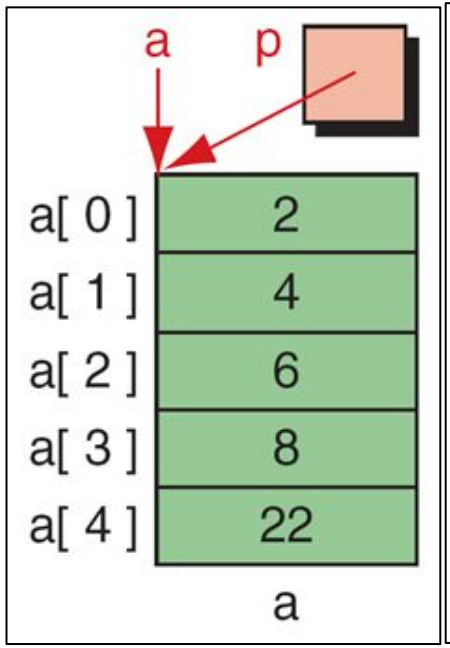
```
2  6  22
2  6  22
```

a[i] is equivalent to *(a+i)

```
#include <stdio.h>
int main() {
   int i, x[6], sum = 0;
   printf("Enter 6 numbers: ");
   for(i = 0; i < 6; ++i) {
      scanf("%d", x+i);
      sum += *(x+i);
   }
    printf("Sum = %d", sum);
return 0;
}
```

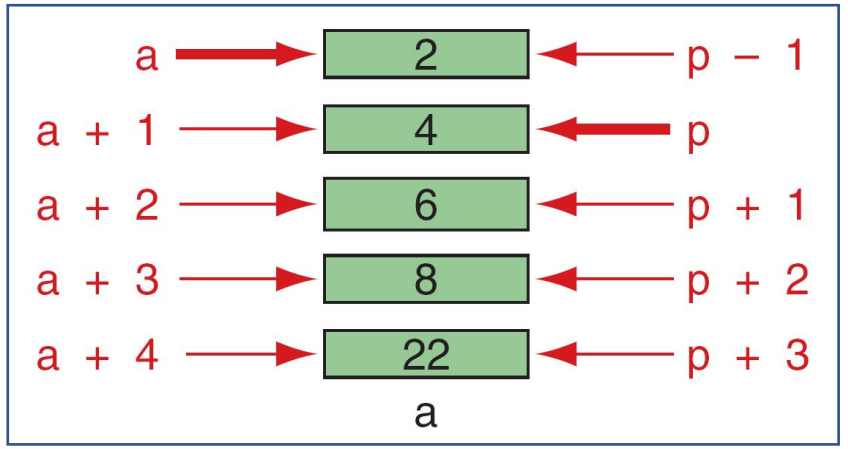
```
Enter 6 numbers:
6
-4
5
3
5
8
Sum = 23
```

# Array names as Pointers




```
#include <stdio.h>
int main(){
    int a[5] = {2, 4, 6, 8, 22};
    int *p = a;
    printf("%d %d %d", a[0], a[2], a[4]);
    printf("\n%d %d %d", *(p), *(p+2), *(p+4));
return 0;
}
```

```
2  6  22
2  6  22
```




```
p = &a[1];

printf("%d %d %d", a[0], a[2], a[4]);    printf("\n%d %d %d", *(p), *(p+2), *(p+4));
```

```
2  6  22
4  8  32765
```

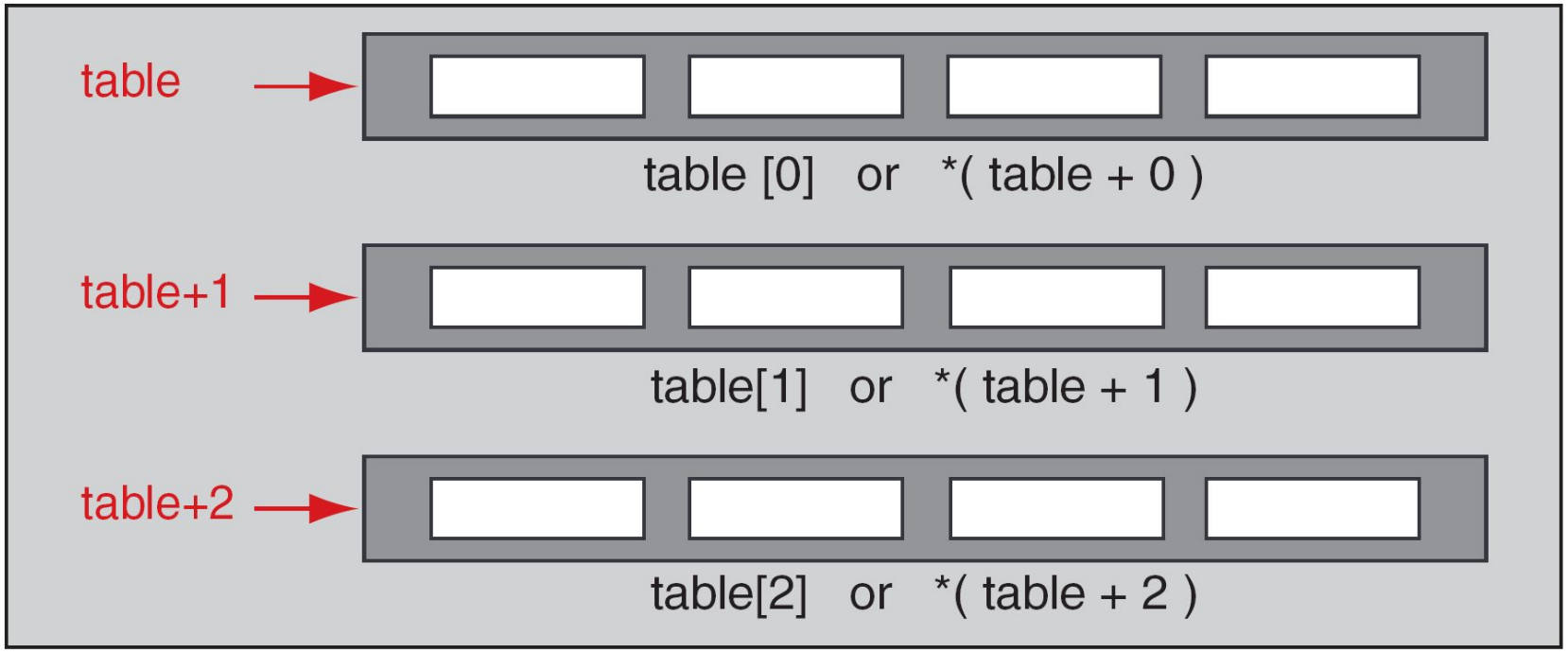
# Pointer Arithmetic: Example1

```
#include <stdio.h>
#define SIZE 5
int main(){
    int arr[SIZE] = {10, 20, 30, 40, 50};
    int *ptr = &arr[0];
    printf("Accessing array elements using pointer \n");
    while(ptr < &arr[SIZE])    {
        printf("%d \n", *ptr);
        ptr++;
     }
    return 0;
}
```

```
Accessing array elements using pointer

10
20
30
40
50
```

# Pointers to Two-dimensional arrays

int table[3][4];




*(arr + i) + j points to the base address of jth element of ith 1-D array

And *( * ( arr + i ) + j ) will point to the value stored there.

# Example

```
#include<stdio.h>
int main(){
   int arr[3][4] = {   {11,22,33,44},
                       {55,66,77,88},
                       {11,66,77,44}      };
   int i, j;
   for(i = 0; i < 3; i++)    {
      printf("Address of %d th array %u \n",i , *(arr + i));
      for(j = 0; j < 4; j++)     {
printf("arr[%d][%d]=%d\n", i, j, *( *(arr + i) + j) );
     }  printf("\n\n");    }    return 0;
}
```

```
Address of 0 th array 826490688
arr[0][0]=11
arr[0][1]=22
arr[0][2]=33
arr[0][3]=44
Address of 1 th array 826490704
arr[1][0]=55
arr[1][1]=66
arr[1][2]=77
arr[1][3]=88
Address of 2 th array 826490720
arr[2][0]=11
arr[2][1]=66
arr[2][2]=77
arr[2][3]=44
```

# Dynamic Memory Allocation in C