ISM
We inspire you to learn

ARRAY OF POINTERS AND POINTER TO AN ARRAY.

Array of pointers:

Up to now we saw the int, char, float arrays. Now we are going to make a look on array of pointers.

Means if an array storing collection of addresses that array is called array of pointers.

Sample program:

programs\pointers\arrofpoi.c

IMPORTANT NOTE

ISM
We inspire you to learn

- ✓remember the difference between the notations*p[3] and (*p)[3]
- ✓Since * has a lower precedence than []
- ✓ *p[3] declares p as an array of 3 pointers while
- ✓ (*p)[3] declares p as a pointer to an array of three elements

int *a[10];
Declares and allocates an **array of pointers** to int. Each element must be dereference individually.

int (*a)[10];
Declares (without allocating) a **pointer to an array** of int(s). The pointer to the array must be dereference to access the value of each element.

int a[10];
Declares and allocates an array of int(s).

SAMPLE PROGRAM:

[programs\pointers\poi2arr12.c](programs\pointers\poi2arr12.c)

```
size of array of 2 (int *) a=8
size of ptr to an array of 2 (int) b=4
size of array of 2 (int) c=8
address of int a[0]=bfa5dc4c value at address *a[0]=1
address of int a[1]=bfa5dc48 value at address *a[1]=2
pointer c=bfa5dc50 value (same as c[0]) *c=1
pointer &c[0]=bfa5dc50 value c[0]=1
pointer &c[1]=bfa5dc54 value c[1]=2
pointer b=bfa5dc50 value is address of c *b=bfa5dc50
pointer *b+0=bfa5dc50 value *(*b+0)=1
pointer *b+1=bfa5dc54 value *(*b+1)=2
```

We inspire you to learn

STRINGS with POINTERS

C [Level1] : pointer and strings

•We can’t assign a string to another string.

•We can assign a character pointer to another character pointer .

•We can’t initialize to another set of characters to a variable.

```
char str[]=“hello”;
char *p=“hello”;
char str1[];
char *q;
str1=str;        //wrong
q=p;             //accepts
str=“hai”;       //wrong
p=“hai   ;       //accepts
SSS
```

C [Level1] : Standard string library

<u>Int strlen(string) :</u>

•It counts the no of characters in a string.

•Base address of string will be passed to the strlen function.

•It doesn't count '\0'.

```
Ex: int i=strlen(str);
          int i=strlen("hello");
```

<u>strcpy(target_string,source_string) :</u>

•This function copies the content in the source string into the target string.

•Base address of source and target address will be passed to the function.

•It copies till ‘\0’ reaches in the source string.

C [Level1] : Standard string library

<u>Int strcmp(string1,string2) :</u>

•This function is used to compare the two strings.

•It returns 0 if two strings are equal.

•It returns the numeric difference between the ASCII value of the first non-matching pair of characters.

C [Level1] : 2-D array of characters

Declaration syntax:

Char name[10][20]={“ism”,”tech”,”hyd”};

```
#include<stdio.h>
main(){
        char names[10][20];
        int i,n;
        printf("\n how many strings you are going to enter::");
        scanf("%d",&n);
        printf("\n enter strings::");
        for(i=0;i<n;i++)
        scanf("%s",names[i]);
        printf("\n entered strings are as follows::\n");
        for(i=0;i<n;i++)
        printf("%s::::%u\n",names[i],&names[i]);
}
```

C [Level1] : 2-D array of characters

Out put:

```
how many strings you are going to enter::4

enter strings::ism
indian
tech
hyd

 entered strings are as follows::
ism::::3221180728
indian::::3221180748
tech::::3221180768
hyd::::3221180788
```

****Note:-** **See the memory allocations.**

C [Level1] : Array of pointers to string

ISM
We inspire you to learn

Declaration syntax:

Char *names[]={“ism”,”tech”,”hyd”};

```
#include<stdio.h>
main()
{
        char *names[]={"ism","indian","hyd","bang"};
        int i;
        printf("\n strings are::");
        for(i=0;i<4;i++)
        printf("%s:::::%u\n",names[i],&(names[i]));
}
```

C [Level1] : Array of pointers to string

Out put:

```
strings are::ism:::::3217380944
indian:::::3217380948
hyd:::::3217380952
bang:::::3217380956
```

Advantages:

•We can make manipulations on strings easily.

•We can reduce the memory wastage.

•Fast accessing is possible.

C [Level1] : Limitations of array of pointers to string

•We can initialize the strings at the place where we are going to declare an array.

•We can’t receive the strings from the keyboard.

•We are declares to array, it is containing garbage values and it won’t be passed to scanf().

Solution for this problem is dynamic memory allocation