C Programming
Day3

by Manjiri Deshpande

Dr. D. Y. Patil Pratishthan's
Institute for Advanced Computing and Software Development

What is a string?

- One dimensional array of characters is called a string
- Synatx:-
- char string name[maxsize];

- Every string terminates with a ‘\0’ character (ASCII Value is 0);

- Char name[15]=“IACSD”

Accept Strings

- ***<u>Using Scanf():-</u>***

%c Specifier-’\0’ character has to be explicitly appended.

E.g.

For(i=0;i<15;i++)

Scanf(“%c”,&name[i]);

Name[i]=‘\0’

Using %s Specifier:-stop accepting when a white space character is entered.

Using gets():-

Accepts string with spaces and there is no need to append ‘\0’

String Functions

- There are many in built string- handling functions declared in ***string.h*** file
- Common ones are

Functions	Purpose
Strlen()	Returns the length of string
Strcpy()	Copies one string into another string
Strcat()	Concatenates two strings
Strcmp()	Compares two string
Strrev()	Reverses the string

Strlen()

- Return number of charcter in a string.
- Exclude null character.
- Int sstrlen(const char*);

- Implementing sstrlen()function
- Int sstrlen(char s[])
- {
- Int i=0,len=0;
- While(s[i]!='\0')
- {
- i++;
- Len++;
- }
- Return len;
- }

Strcpy()

- Assigning the value of one string to other with an assignment operator is not possible.
- Strcpy() function copies the source string to target string;
- char * strcpy(char *target, char*src);
- char src[]="iacsd";
- char target[20];
- strcpy(target,src);

Problem Scenario

- Consider an example
- To store 5 fixed deposit amounts each for 10 customers.
- Choices could be:
- Declare 50 Variables.
- An array with 50 elements.
- 10 arrays with 5 elements each.
- What if the number of fixed deposits or number of customer increases?

Two Dimensional Array

- Considering the same example2 fixed deposit amounts 10 customers each, can be represented using a two dimensional array .Two dimensional array is also called an array of arrays. It is represented as shown below.
- Synatx :-int acc[10][2];[row size][col]
- Data type arrayname[rowsize][colsize];
- acc is an array of 10 elements.
- Each element of array acc is an array of 2 integers.
- Similarly acc[0],acc[1]….so on are the elements of an array.
- acc[0] is itself an array,its first elements is acc[0][0] ,second element[0][1] and so on….

2D Array:-

- 10 one dimensional arrays=number of rows in a 2D array.
- int acc[10][2];

```
int acc[10][2];
int acc[2][2]={1001,20000,1002,10000};
OR
  int acc[2][2]={{1001,20000},{1002,10000}};
OR
  int acc[][2]={{1001,20000},{1002,10000}};
```

Different ways of 2D array with a list of initial values enclosed in braces

Accessing Elements of 2D array

- Elements of a 2D array can be accessed by suing two notations:
- 1.Subscript notation
- 2.Pointer notation
- To Access any element in a2D array by using subscript notation is similar to accessing elements in a 1D array.
- E.g.The 2nd element in the 3rd row can be accessed as acc[2][1].
- The same element can be accessed using the pointer notation as(*(*(acc+2)+1)

Pointer Notation of 2D Array

- ***The same element can be accessed using the pointer notation as(*(*(acc+2)+1)***

--*(acc+2)→ gives the address of the 3rd row

--*(acc+2)+1→gives address of 2nd element of 3rd row

--*(*(acc+2)+1)→gives value stored at the 2nd element in the 3rd row

Accepting and displaying a 2D Array

- Accept 2D Array

```
for(i=0i<rows;i++)
for(j=0;j<cols;j++)
scanf(“%d”,&acc[i][j]);
```

Display 2D Array

```
for(i=0i<rows;i++)
for(j=0;j<cols;j++)
printf(“%d\t”,acc[i][j]);
```

Dynamic Memory Allocation

- Size of array is fixed
- Memory may be insufficient or may be wasted.
- ***<u>Consider the following scenarios</u>:***
- ***Number of elements for 1D array is not known.***
- ***2D array***
- ***Row size is known but column size is not known.***
- ***Neither row size nor column size is known.***
- ***DMA allows memory to be allocated at run time***.

Functions for Dynamic Memory Allocations

--malloc():-syntax

```
void *malloc(size_t *number of element);
```

It returns a pointer of type void * to the starting location of the block of memory allocated.

If memory allocations fails, a NULL pointer is returned

Include<stdlib.h>

```
int *acc,num;
acc=(int *)malloc(num*sizeof(int));
```

Functions for Dynamic Memory Allocations

--calloc():-

*void *calloc(size_t elements,Size _t sz);*

It is similar to malloc.

It allocates space for an array of elements,each of which occupies sz bytes storage.

The space of each element is initialized to binary Zeros.

```
int *acc,num;
acc=(int *)calloc(100,sizeof(int));
```

Functions for Dynamic Memory Allocations

- --realloc():-This function is used to append new memory to the existing memory block .If 20 bytes of memory have been assigned to *ptr using malloc and later 20 more bytes are required to add to ptr,realloc() adds 20 bytes to the existing 20 bytes.

```
Void * realloc(void *ptr,size_ts);
```

```
ptr=realloc(ptr,(n+5)*sizeof(int));
```

void pointer in C

- ***Till now, we have studied that the address assigned to a pointer should be of the same type as specified in the pointer declaration***. For example, if we declare the int pointer, then this int pointer cannot point to the float variable or some other type of variable, i.e., it can point to only int type variable. To overcome this problem, we use a pointer to void. A pointer to void means a generic pointer that can point to any data type. We can assign the address of any data type to the void pointer, and a void pointer can be assigned to any type of the pointer without performing any explicit typecasting.

What is a Null Pointer?

- A Null Pointer is a pointer that does not point to any memory location. It stores the base address of the segment. The null pointer basically stores the Null value while void is the type of the pointer.
- A null pointer is a special reserved value which is defined in a **stddef** header file. Here, Null means that the pointer is referring to the 0th memory location.
- If we do not have any address which is to be assigned to the pointer, then it is known as a null pointer. When a NULL value is assigned to the pointer, then it is considered as a **Null pointer**.

Array of Pointer

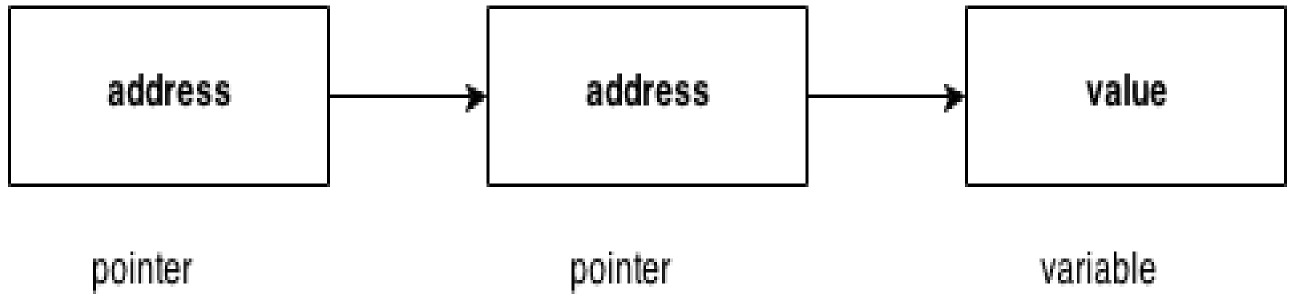
- In case of 2D array ,the size of a column is not known at compile time,only rows size is known,

```
int *acc[rowsize];
```

- [] operator has a higher precedence than * operator,hence array of pointer.
- Acc is an array of pointer on stack.
- Number of column is accepted from user.
- Necessary memory is allocated on heap.

Pointer to Pointer

- In C, we can also define a pointer to store the address of another pointer. Such pointer is known as pointer to pointer. The first pointer is used to store the address of a variable whereas the second pointer is used to store the address of the first pointer. Let's understand it by the diagram given below.


address
address
value
pointer
pointer
variable