STRINGS

STRINGS

- C compiler supports large number of string handling library functions.
- string.h is to be included whenever library function is used.
- Strings in C are represented by arrays of characters.
- The end of the string is marked with a special character, the null character.

CONTD.

- The group of characters, digits, and symbols enclosed within quotation marks are called as strings
- The string is always declared as character arrays
- Declaration of a one-dimensional array:
- <data type> <arrayname>[<SIZE>]
- The array elements are all values of the type <type>
- The size of the array is indicated by SIZE
- Example:
  - char name[ ]={'I','N','D','I','A‘,‘\0'}
  - char name[ ]=”INDIA”;

STRING FUNCTIONS

- *strcmp(str1,str2):* Compares two strings and returns an integer indicating the difference between the strings
- *strcat(dest,src):* Concatenates src to the end of dest
- *strcpy(dest,src):* Copies src string to dest
- *strlen(str):* Returns the length of the string (doesn't count NULL character)
- *strlwr(str):* Convert the string into lower case
- *strupr(str):* Convert the string into upper case
- *strncpy(dest,src,n):* copies upto *n* characters
- *strncmp(str1,str2,n):* compares *n* characters
- *strncat(dest,src,n):* concatenates *n* characters

CONTD.

- *memcpy(dest,src,n):* copies block of *n* bytes
- *memcmp(dest,src,n):* compares first *n* bytes
- *memset(str,ch,n):* sets first *n* bytes of str to ch
- *strchr(str,ch):* scans for the first occurrence of ch
- *strset(str,ch):* sets all characters of str to ch
- *strnset(str,ch,n):* sets first *n* characters of str to ch
- *strrev(str):* reverses the string
- *strstr(str1,str2):* scans str1 for the first occurrence of substring str2 and returns a pointer
- *itoa(val,str,radix):* converts val to str using base as radix
- *atoi(str):* converts string of digits to integer and returns it

TWO DIMENSIONAL ARRAY OF STRINGS

- A two-dimensional array of strings can be declared as follows:
- <data_type> <string_array_name> [<row_size>][<column_size>];
- char s[5][10] = {“Cow”, “Goat”, “Cat”, “Lion”, “Deer”}

  s[0] = C o w \0

  s[1] = G o a t \0

  s[2] = C a t \0

  s[3] = L i o n \0

  s[4] = T i g e r \0

  Every row is a string, i.e., s[i] is a string

MANIPULATING STRING – 2D ARRAYS

- **Code to scan and print an individual string of an array of strings**

```
main()
{
        char s[10][30];
        int i;
        for (i=0;i<10;i++)
                scanf("%s", s[i]);
        for (i=0;i<10;i++)
                printf("\n %s", s[i]);
    }
```

MANIPULATING STRING - 2D ARRAYS

```
main()
    {
            int i,j,n; char a[20][20], temp[10];
            printf(“\n How many strings: ”);
            scanf(“%d”, &n);
            for (i=0;i<n;i++)
                    scanf(“%s”, a[i]);
            for (i=0; i<n-1;i++)
            {
            for(j=i+1; j<n; ++j)
                if (strcmp(a[i],a[j])>0)
                {
                    strcpy(temp,a[i]);
                    strcpy(a[i],a[j]);
                    strcpy(a[j],temp);
                 }
            }
            for (i=0; i<n; i++)
                    printf(“\n %s”, a[i]);
    }
```