Strings

What are strings?

- Like group of integers can be stored in an integer array group of characters are also known as strings
- It is terminated by ‘\0’ null character
- Occupies one byte of memory

String functions

- strlen
- strlwr
- strupr
- strcat
- strncat
- strcpy
- strncpy
- strcmp

and so on..

Different const

- char *p
- const char *p
- char const *p
- char * const p
- const char * const p

Two dimensional array of characters

- Syntax

```
char multi[6][10] ={
                        “rishabh”,
                        ”nilesh”,
                        “nisarg”,
                        “yash”,
                        “jignesh”
                    }
```

Array of pointers to strings

- Syntax:

```
char *names [ ] = {
                    “rishabh”,
                    ”nilesh”,
                    “anand”,
                    “yash”,
                    “jignesh”
                  }
```

Note: This is better than two dimensional array

What is the problem with this code?

```
main( )
{
char *names[6] ;
int i ;

for ( i = 0 ; i <= 5 ; i++ )
{
printf ( "\nEnter name " ) ;
scanf ( "%s", names[i] ) ;
}
}
```

Problem???

Solution

- malloc to the rescue

```
char *names[6] ;
char n[50] ;
int len, i ;
char *p ;
for ( i = 0 ; i <= 5 ; i++ )
{
printf ( "\nEnter name " ) ;
scanf ( "%s", n ) ;
len = strlen ( n ) ;
p = malloc ( len + 1 ) ;
strcpy ( p, n ) ;
names[i] = p ;
}
```

Summary

- A string is nothing but an array of characters terminated by '\0'.
- Being an array, all the characters of a string are stored in contiguous memory locations.
- Though **scanf( )** can be used to receive multi-word strings, **gets( )** can do the same job in a cleaner way.
- Both **printf( )** and **puts( )** can handle multi-word strings.
- Strings can be operated upon using several standard library functions like **strlen( )**, **strcpy( )**, **strcat( )** and **strcmp( )** which can manipulate strings. More importantly we imitated some of these functions to learn how these standard library functions are written.
- Though in principle a 2-D array can be used to handle several strings, in practice an array of pointers to strings is preferred since it takes less space and is efficient in processing strings.
- **malloc( )** function can be used to allocate space in memory on the fly during execution of the program.

Questions

- Will this compile?

```
char a[]="Sunstroke";
a="Coldwave";
```

- Will this compile?

```
char *p="Coldwave";
p="Sunstroke";
```

- char str[]="sales\0man\0"

printf("%d",sizeof(str));

- char str[]="sales\0man\0"

printf("%d",strlen(str));

- printf(“%c”,”abcdefgh”[4]);
- printf(“%d %d %d”, sizeof(3.0f),sizeof(“3”),sizeof(3.0));
- static char s[]= “Hello”;
- printf (*(s + strlen(s)))

Answers

- No
- Yes
- 11
- 5
- e
- 4 2 8
- 0

Contact Info

- trainers@finaldesk.com
- rishabh@finaldesk.com
- nilesh@finaldesk.com
- jignesh@finaldesk.com
- yash@finaldesk.com
- anand@finaldesk.com