module 3 (contd.....)

Strings :- (part 2)

Declaration & Initializatn of string handling functions:

- strlen()
- strcpy()
- strcat()
- strrev()
- strcmp()
- strncmp()
- strncpy()
- strlwr()
- strupr()

} very important

*** lab program 7 (string concatenation)

String :- Sequence or array of characters which ends with null character.

Storage representatn of string is classified into 2 types :-

1) Fixed length string
2) variable length string

1) Fixed length string :-

* The length of string is pre-defined.
* If we try to add more characters than the specified size, then string

will be truncated.

Disadvantages :-

If size is 100, if we store 5 character then remaining 95 fields will be wasted.

2) Variable length string

No predefined length.
storage of string can be varied.

2 mechanism to indicate end of string

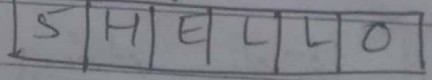
1) Length controlled string
2) Delimited string

1) Length controlled string

In this, size of string is stored as a part string itself.

Disadvantage :- Length can't exceed 255

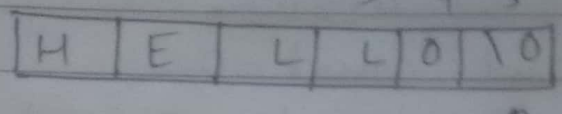
Eg:-



2) Delimited string

String ends with delimiter (null character)

Eg:-



↑ delimiter.

1) * Declare & initialize string:
* How to read string:
↳ gets(), puts(), scanf(), printf().

1) Declaration:-

```
char a[10];
↓    ↓   ↓
type string- size
     name
```

Initialization:-

1) char a[5] = {'H', 'E', 'L', 'L', 'O'};

In this, character by character initializatn is done.

2) char a[6] = "HELLO";
↳ size is 6 including null character.

* WAP to accept string & count no. of characters in it without using builtin function.

```
void main()
{
```

```
char a[10];
int count=0, i;
printf("Enter string \n");
scanf("%s", a);
for(i=0; a[i]!='\0'; i++)
  count = count+1;
printf("%s contains %d characters",
        a, count);
}
```

* WAP to find reverse of a given string without using builtin function.

```
void main()
{
  int i, length=0;
  char s[10];
  printf("Enter string \n");
  gets(s);
  while(s[i]!='\0')
  {
    length = length+1;
    i++;
  }
  printf("Reversed string is \n");
  for(i=length-1; i>=0; i--)
    printf("%c", s[i]);
  getch();
}
```

*** String handling functions (In built):-

1) strlen(str):- To find length of string.

```
#include<stdio.h>
#include<string.h>
void main()
{
    char str[10] = "Hello";
    clrscr();
    printf("length = %d", strlen(str));
}
```

o/p = 5

2) strcpy(dest, src) :- String is copied from source to destination.

```
#include<string.h>
void main()
{
    char src[100], dest[100];
    gets(src);
    strcpy(dest, src);
    puts(dest);
    getch();
}
```

o/p: src = Hello
dest = Hello

3) strncpy(dest, src, n)

specified number of characters are copied from source to destination.

Eg:-
```
#include<string.h>
void main()
{
    char src[100], dest[100];
    gets(src);
    strncpy(dest, src, 5);
    puts(dest);
}
```

o/p: COMPUTER → src
COMPU → Dest

In this eg, only 5 characters are copied from src to dest.

4) strcat(str1, str2) :- (concatenation)

(or) string2 will be joined with string1. string2 will be copied to string1.

```
#include<string.h>
void main()
{
    char s1[100], s2[100];
    printf("Enter string 1\n");
    scanf("%s", s1);
    printf("Enter string 2\n");
    scanf("%s", s2);
    strcat(s1, s2); printf("%s", s1);
    getch();
}
```

o/p:- string 1 = Hello
string 2 = Hi
string 1 = HelloHi

5) strncat (s1, s2, n) :-

only specified no. of characters from s2 will be joined with s1.

```
#include <string.h>
void main()
{
  char s1[10] = "Hello", s2[10] = "hello";
  strncat (s1, s2, 2);
  printf ("%s", s1);
}
```

o/p:- Hellohe.

6) strrev (str) :-

To reverse a given string.

```
#include<string.h>
void main()
{
  char s[10];
  gets(s);
  strrev(s);
  puts(s);
}
```

o/p:- s - Hello
reverse - olleH.

7) strlwr(str):-
converts string to lowercase letters

```
#include<string.h>
void main()
{
    char s[10];
    gets(s);
    strlwr(s);
    puts(s);
}
```

I/P: HELLO
O/P: hello

8) strupr(str):-
converts string to uppercase string.

```
#include<stdio.h>
#include<string.h>
void main()
{
    char s[100];
    gets(s);
    strupr(s);
    puts(s);
}
```

I/P: hello
O/P: HELLO

9) strcmp(s1, s2):-

* Compares 2 strings.
* comparision starts with first character of each string, continues till the corresponding characters differs or end of character is reached.

→ If 2 strings are equal returns 0.
→ S1 > S2 → returns +ve.
→ S1 < S2 → returns -ve.

It returns values based on ASCII values of letters subtraction.

```
#include<string.h>
void main()
{
    char s1[100], s2[100];
    int ch;
    scanf("%s %s", s1, s2);
    ch = strcmp(s1, s2);
    printf("%d", ch);
}
```

10) strncmp(s1, s2, n);-

only specified number of characters will be compared in s1 & s2.

```
#include<string.h>
void main()
{
    char s1[100], s2[100];
    int ch;
    scanf("%s%s", s1, s2);
    ch = strncmp(s1, s2, 4);
    printf("%d", ch);
}
```