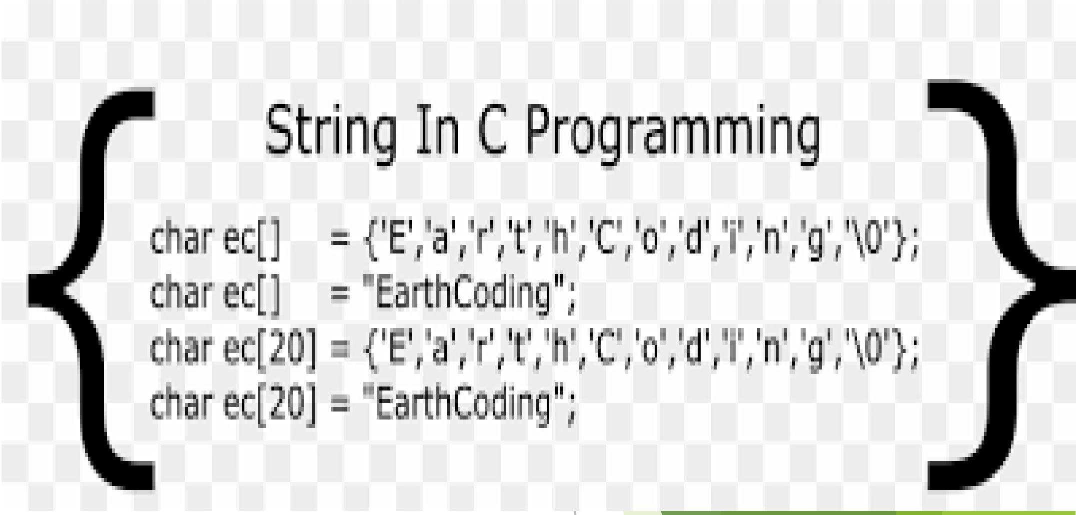
# ਪਾਠ-2
# ਸਟਰਿੰਗ ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ

ijIQa

Made by
Neetu Arora
Kiran Bala
GSSS FAIZULLAPUR(FGS)

# ਸਟਰਿੰਗ

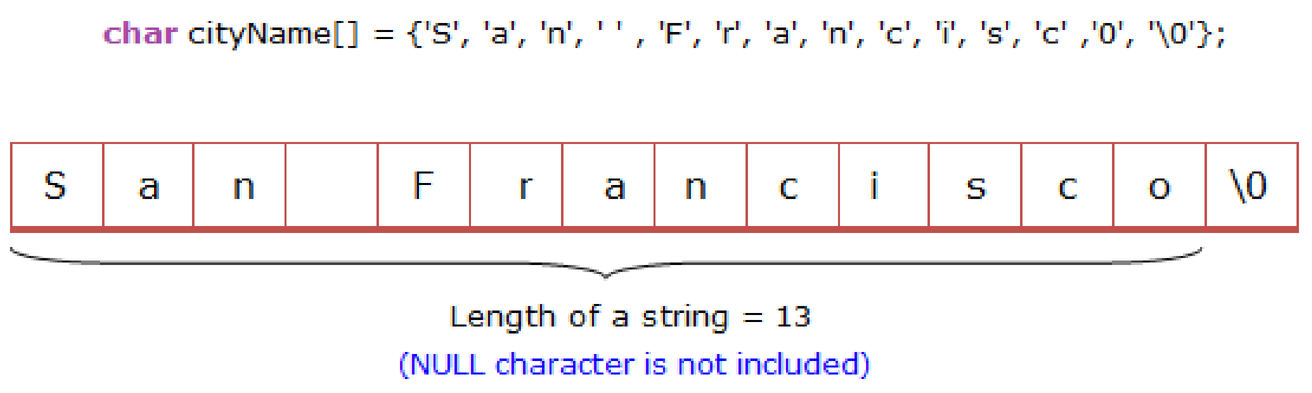



1. ਸਟਰਿੰਗ ਚਿੰਨ੍ਹਾ ਦਾ ਸਾਰ ਹੁੰਦਾ ਹੈ ।
2. ਇਹ ਦੋਹਰੇ ਕਾਮਿਆਂ ਵਿਚ ਲਿਖਿਆ ਜਾਂਦਾ ਹੈ ।
3. ਇਸ ਦਾ ਅੰਤ ਨੱਲ ਚਿੰਨ੍ਹ ਨਾਲ ਜਾਂ ਖਾਲੀ ਚਿੰਨ੍ਹ ਨਾਲ ਹੁੰਦਾ ਹੈ।
4. ਸੀ ਭਾਸ਼ਾ ਵਿਚ ਸਟਰਿੰਗ ਸਟੋਰ ਕਰਨ ਲਈ ਕੈਰ ਐਰੇ Build-in ਫੰਕਸ਼ਨ ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ।

# ਸਟਰਿੰਗ ਦੀ ਉਦਾਹਰਣ

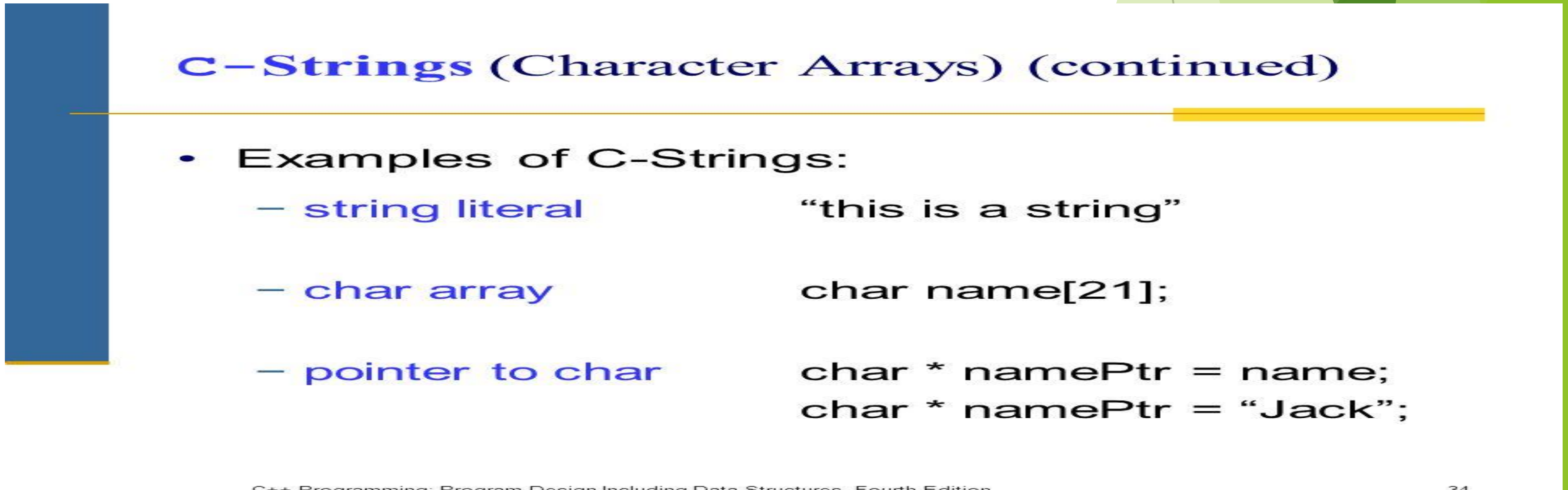
- ਕੁਝ ਆਮ ਵਰਤੇ ਜਾਨ ਵਾਲੇ ਸਟਰਿੰਗ ਫੰਕਸ਼ਨਾ ਦੀਆਂ ਉਦਾਹਰਣਾਂ

- 1 strcpy()
- 2 strcat()
- 3 strcmp()
- 4 strlwr()
- 5 strupr()
- 6 strlen()



# CHAR ARRAY

- ਸੀ ਭਾਸ਼ਾ ਵਿਚ ਐਰੇ ਕੈਰ ਦਾ ਵਿਸ਼ੇਸ਼ ਮਹੱਤਵ ਹੈ।
- ਸੀ  ਭਾਸ਼ਾ ਵਿਚ ਸਟਰਿੰਗ ਭਾਸ਼ਾ ਸਟੋਰ ਕਰਨ ਲਈ ਕੋਈ ਵੀ ਡਾਟਾ ਟਾਈਪ ਮੌਜੂਦ ਨਹੀ ਹੈ ਇਸ ਲਈ ਸਟਰਿੰਗ ਨੂੰ ਸਟੋਰ ਕਰਨ ਲਈ ਕੈਰ ਐਰੇ ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ



# CHAR ARRAY ਉਦਾਹਰਣ

- 
- Char name [20]
- 
- ਇਸ ਉਦਾਹਰਣ ਵਿਚ name ਨਾਂ ਦਾ ਇਕ char array ਡਿਕਲੇਰ ਕੀਤਾ ਹੈ ਜੋ ਕਿ 20 ਅੱਖਰਾਂ ਤੱਕ ਦਾ ਕੋਈ ਸਟਰਿੰਗ ਸਟੋਰ ਕਰ ਸਕਦਾ।

# ਸਟਰਿੰਗ ਵੇਰੀਐਬਲਸ ਦੀ ਡਿਕਲੇਅਰੇਸ਼ਨ

- ਸੀ ਵਿਚ  ਅਸੀ ਸਟਰਿੰਗ ਵੇਰੀਐਬਲਸ ਲਈ ਸਟਰਿੰਗ ਦੀ ਵਰਤੋਂ ਕਰਦੇ ਹਾਂ|
- ਸਟਰਿੰਗ ਵੇਰੀਐਬਲਸ ਡਿਕਲੇਅਰ ਕਰਨ ਦਾ ਆਮ  ਰੂਪ ਹੈ
- Data-type string- name [size]

# ਸਟਰਿੰਗ ਵੇਰੀਐਬਲਸ ਇਨੀਸ਼ਿਲਾਈਜੇਸ਼ਨ

ਕੈਰ ਐਰੇ ਨੂੰ ਡਿਕਲੇਅਰ ਕਰਨ ਤੋਂ ਬਾਅਦ ਇਨੀਸ਼ਿਲਾਈਜ ਕੀਤਾ ਜਾਂਦਾ
ਹੈ

ਕੈਰ ਐਰੇ ਨੂੰ ਇਨੀਸ਼ਿਲਾਈਜ ਕਰਨ ਦੇ 2 ਤਰੀਕੇ ਹਨ।

1 char name[10]={‘R’,’a’,’m’,’a’’\0’};

2 char days [10]={“MONDAY”};

# ਸਟਰਿੰਗ ਵੇਰੀਐਬਲਸ ਦੀ ਡਿਕਲੇਅਰੇਸ਼ਨ

```
char string_variable_name [array_size];
```

The classic string declaration can be done as follow:

```
char string_name[string_length] = "string";
```

# String ਇਨੀਸ਼ਿਲਾਈਜੇਸ਼ਨ ਲਈ ਪਰੋਗ੍ਰਾਮ

This example program illustrates initialization of an array of C strings.

```
#include <stdio.h>

const char * array[] = {
    "First entry",
    "Second entry",
    "Third entry",
};

#define n_array (sizeof (array

int main ()
{
    int i;

    for (i = 0; i < n_array; i
        printf ("%d: %s\n", i,
    }
    return 0;
}
```

It prints out the following:

```
0: First entry
1: Second entry
2: Third entry
```

# ਸਟਰਿੰਗ ਪੜ੍ਹਨਾ

- ਇਨਪੁਟ ਸਟੇਟਮੈਂਟਸ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਸਟਰਿੰਗ ਨੂੰ ਪੜਿਆ ਜਾ ਸਕਦਾ ਹੈ।
- ਸਟਰਿੰਗ ਪੜ੍ਹਨਾ(Reading Strings)
- 1)scanf() ਦੇ ਨਾਲ %s format specifications
- 2) gets() function
- 3)getchar() function

# STRING ਲਿਖਣਾ

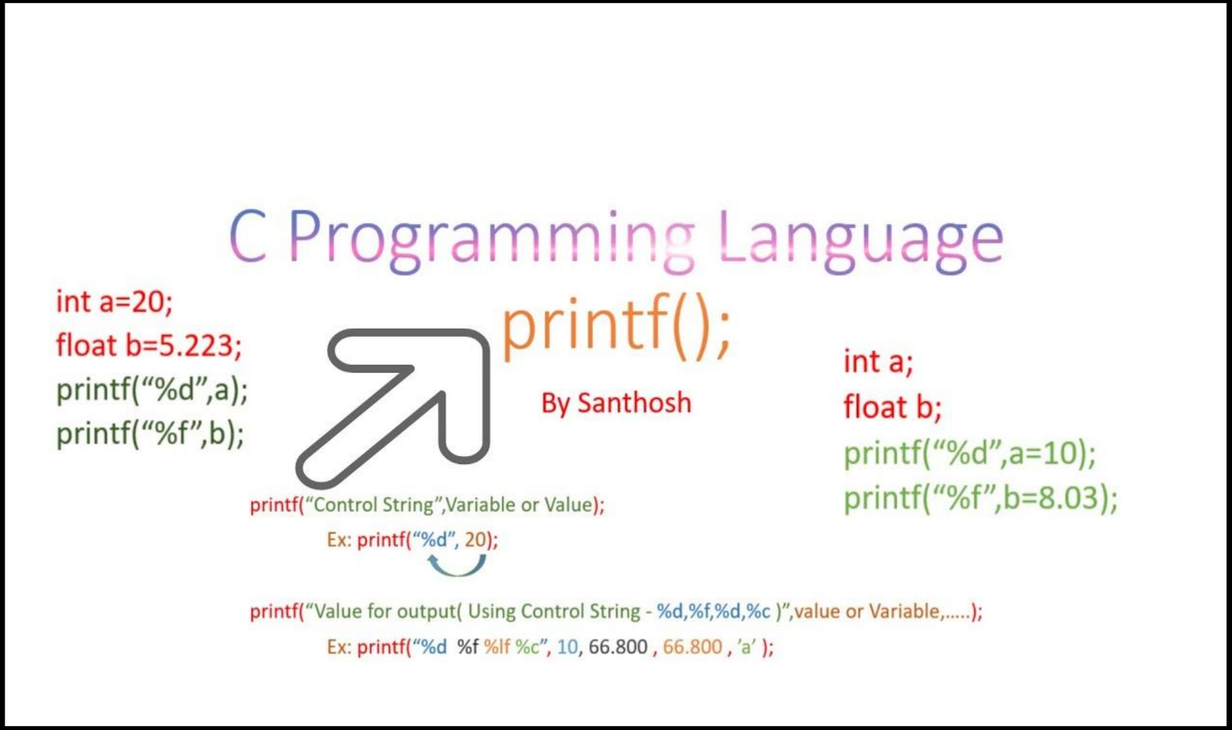
- ਆਊਟਪੁਟ ਸਟੇਟਮੈਂਟਸ ਦੀ ਵਰਤੋਂ ਕਰਕੇ ਸਟਰਿੰਗ ਨੂੰ ਲਿਖਿਆ ਸਕਦਾ ਹੈ
- ਸਟਰਿੰਗ ਲਿਖਣਾ(Writing Strings)
- 1) printf() ਦੇ ਨਾਲ %s format specifications
- 2) puts()function
- 3) putchar() function

| Format String | Meaning |
| --- | --- |
| %d | Scan or print an integer as signed decimal number |
| %f | Scan or print a floating point number |
| %c | To scan or print a character |
| %s | To scan or print a character string. The scanning ends at whitespace. |

# Scanf() ਉਤੇ printf()

## Program to find quotient and remainder using printf() and scan(f) function

```
#include <stdio.h>

int main()
{
 int dividend, divisor;
 int quotient, remainder;

 printf("Enter dividend: ");
 scanf("%d",&dividend);
 printf("Enter divisor: ");
 scanf("%d",&divisor);

 quotient= dividend/divisor;
 remainder= dividend%divisor;

 printf("quotient: %d, remainder:
%d\n",quotient,remainder);

 return 0;
}
```



Output

```
First run:
Enter dividend: 10
Enter divisor: 3
quotient: 3, remainder: 1

Second run:
Enter dividend: 10
Enter divisor: 2
quotient: 5, remainder: 0

Third run:
Enter dividend: 10
Enter divisor: 100
quotient: 0, remainder: 10
```

# Gets FUNCTION

- Scanf ਦੀ ਵਰਤੋਂ ਨਾਲ ਇਨਪੁਟ ਕਰਨ ਵਿਚ ਸਮਸਿਆ ਇਹ ਹੈ ਕਿ ਇਹ ਸਟੇਟਮੈਂਟ ਵਿਚ ਇਕ ਖਾਲੀ ਸਥਾਨ ਮਿਲਨ ਤੇ
- ਇਸ ਨੂੰ ਖਤਮ ਕਰ ਦਿੰਦੀ ਹੈ।
- ਮੰਨ ਲਿਉ ਕਿ ਅਸੀ ਸਟਰਿੰਗ “new York” ਟਾਈਪ ਕੀਤਾ ਹੈ ਤਾਂ ਕੇਵਲ ਸਟਰਿੰਗ “new” ਹੀ ਪੜ੍ਹੀ ਜਾਵੇਗੀ ਕਿਉਂਕਿ ਜੋ
- ਸ਼ਬਦ “new” ਤੋਂ ਬਾਅਦ ਖਾਲੀ ਸਥਾਨ ਹੈ ਇਸ ਲਈ ਇਹ ਸਟਰਿੰਗ ਨੂੰ ਖਤਮ ਕਰ ਦਵੇਗੀ।
- Scanf ਬਹੁਸ਼ਬਦੀ ਸਟਰਿੰਗ ਪ੍ਰਾਪਤ ਕਰਨ ਦੇ ਕਾਬਿਲ ਨਹੀਂ ਹੈ ਇਸ ਲਈ ਨਾਮ ਜਿਵੇਂ ਕੀ RAM GOPAL ਗੈਰ ਪ੍ਰਵਾਨ ਯੋਗ ਹੋਵੇਗਾ। ਇਸ ਦੇ ਹਲ ਲਈ gets function ਦੀ ਵਰਤੋਂ ਕੀਤੀ ਜਾਂਦੀ ਹੈ।
- gets function()ਅਤੇ ਇਸ ਦੇ counter part puts() ਦੀ ਵਰਤੋਂ ਹੇਠ ਦਰਸਾਈ ਗਈ ਹੈ।

gets () ਸਟਰਿੰਗ ਰੀਡ ਕਰਨ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ। ਹੈਡਰ ਫਾਈਲ ਵਿਚ ਪ੍ਰਭਾਸ਼ਿਤ ਹੈ

syntax for gets function is:

gets(char_array_variable)

# puts ਫੰਕਸ਼ਨ

Puts() stdio.h ਹੈਡਰ ਫਾਈਲ ਵਿਚ ਪ੍ਰਭਾਸ਼ਿਤ ਹੈ।ਅਤੇ ਆਉਟਪੁਟ ਲਈ ਵਰਤਿਆ ਜਾਂਦਾ ਹੈ

puts(char_array_variable/string)

# Gets() ਅਤੇ put() ਫੰਕਸ਼ਨ ਲਈ ਪ੍ਰੋਗਰਾਮ

```c
#include <stdio.h>

int main(){

  char ch[100];
  printf("Enter any string\n");
  //get the string
  gets(ch);
  printf("\nThe string is - ");
  //print the string
  puts(ch);

  return 0;
}
```

17:10

# Output

Command Prompt

```
C:\Users\Souvik\includehelp>a
Enter the string
Includehelp tutorial

The string is - Includehelp tutorial
C:\Users\Souvik\includehelp>
```

# Getchar() and putchar()

- Getchar(): ਸਟੈਂਡਰ ਇਨਪੁਟ ਡਿਵਾਈਸ ਤੋਂ ਪੜ੍ਹਨ ਲਈ getchar ਫੰਕਸ਼ਨ ਵਰਤੇ ਜਾਂਦੇ ਹਨ।
- Getchar ਦਾ ਰੂਪ ਹੈ:
- Variable name=getchar()

Putchar():ਇਹ ਇਕ ਅਨਫਾਰਮੇਟਡ ਕਰੈਰਟਰ ਆਊਟਪੁਟ ਫੰਕਸ਼ਨ ਹੈ।ਇਸ ਫੰਕਸ਼ਨਦੀ ਵਰਤੋਂ ਅਨਫਾਰਮੇਟਡ ਕਰੈਕਟਰ ਨੂੰ ਕਰੈਰਟਰ ਨੂੰ ਮੋਨੀਟਰ ਸਕਰੀਨ ਤੇ ਦਰਸ਼ਾਉਣ ਲਈ ਕਤਿੀ ਜਾਂਦੀ ਹੈ

Putchar(variable name)

# Getchar() and putchar() progarm

```
#include <stdio.h>
#include <ctype.h>
int main()
{
    char c;
    printf("Enter some character. Enter $ to exit...\n");
    while (c != '$');
    {
        c = getchar();
        printf("\n Entered character is: ");
        putchar(c);
        printf("\n")
    }
    return 0;
}
```

## OUTPUT:

```
Enter some character. Enter $ to exit...
A
Entered character is: A
B
Entered character is: B
$
Entered character is: $
```

# ਸਟਰਿੰਗ ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ

- (ਸੀ) ਭਾਸ਼ਾ String Handling Functions ਪ੍ਰਦਾਨ ਕਰਦੀ ਹੈ, ਜਿਹੜੇ ਹੈਡਰ ਫਾਈਲ (header file) <string.h> ਵਿੱਚ ਪਰਿਭਾਸ਼ਿਤ ਕੀਤੇ ਗਏ ਹਨ।
- ਜਦੋਂ ਵੀਂ ਸਟਰਿੰਗ ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰਨੀ ਹੁੰਦੀ ਹੈ ਤਾਂ ਹੈਡਰ ਫਾਈਲ ਨੂੰ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਸ਼ਾਮਿਲ ਕਰਨਾ ਪੈਂਦਾ ਹੈ।

# ਸਟਰਿੰਗ ਫੰਕਸ਼ਨਾਂ ਦੀ ਲਿਸ਼ਟ :-

- strlen( )
- strcat( )
- strcmp( )
- strupr( )
- strlwr( )
- strrev( )

# strlen( )

ਫੰਕਸ਼ਨ strlen ਦਾ ਪੂਰਾ ਨਾਂ String Length ਹੈ ਇਸ ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਦਿੱਤੇ ਗਏ ਸਟਰਿੰਗ ਵਿੱਚ ਅੱਖਰਾਂ ਦੀ ਗਿਣਤੀ ਪਤਾ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਲੰਬਾਈ ਵਿੱਚ ਕੋਈ ਖਾਲੀ ਚਿੰਨ੍ਹ ਨਹੀਂ ਹੁੰਦਾ ।

ਵਾਕ-ਰਚਨਾ (SYNTAX) :-

N=strlen(“string”);

ਜਿੱਥੇ N ਇੱਕ ਇੰਟੀਜਰ-ਵੇਰੀਏਬਲ ਹੈ ਉਹ ਸਟਰਿੰਗ ਦੀ ਲੰਬਾਈ ਦਾ ਮੁੱਲ ਪ੍ਰਾਪਤ ਕਰਦਾ ਹੈ।

ਮਿਸਾਲ :-

n=strlen(“School”);

strlen ਫੰਕਸ਼ਨ 6 ਵਾਪਿਸ ਕਰੇਗਾ ।

# ਮਿਸਾਲ :- Strlen() ਫੰਕਸ਼ਨ ਵਰਤ ਕੇ ਸਟਰਿੰਗ ਦੀ ਲੰਬਾਈ ਪਤਾ ਕਰਨ ਲਈ ਪ੍ਰੋਗਰਾਮ ।

```
#include<stdio.h>
#include<string.h>
void main( )
{
int n;
n=strlen("School");
printf("Length of the string=%d",n);
getch( );
}
```

Output of the Program (ਪ੍ਰੋਗਰਾਮ  ਦਾ ਪਰਿਣਾਮ ):-

Length of the string=6

# strcat( )

ਫੰਕਸ਼ਨ  strcat ਦਾ ਪੂਰਾ ਨਾਂ String concatenation ਹੈ । ਇਸ ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਦੋ ਸਟਿਰੰਗ ਮੁੱਲਾਂ ਨੂੰ ਇੱਕ ਸਟਰਿੰਗ ਮੁੱਲ  ਵਿੱਚ ਇਕੱਠੇ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ। ਇਹ ਫੰਕਸ਼ਨ ਦੂਜੇ ਸਟਰਿੰਗ ਮੁੱਲ ਨੂੰ ਪਹਿਲੇ ਸਟਰਿੰਗ ਦੇ ਅੰਤ ਵਿੱਚ ਜੋੜ ਦਿੰਦਾ ਹੈ ।

ਵਾਕ-ਰਚਨਾ (SYNTAX) :-

**strcat(string1,string2);**

ਜਿੱਥੇ  string1 ਅਤੇ string2 character array ਹਨ।

ਮਿਸਾਲ (Example) :-

**strcat("Aman","deep");**

## Output:-

Amandeep

# ਮਿਸਾਲ :- ਦੋ ਸਟਰਿੰਗਜ਼ ਨੂੰ ਜੋੜਨ ਲਈ ਪ੍ਰੋਗਰਾਮ ।

```
#include<stdio.h>
#include<string.h>
void main( )
{
printf(strcat("Aman","deep"));
getch( );
}
```

**Output of the Program** (ਪ੍ਰੋਗਰਾਮ ਦਾ ਪਰਿਣਾਮ ) :-

Amandeep

# strcmp( )

ਫੰਕਸ਼ਨ strcmp ਦਾ ਪੂਰਾ ਨਾਂ String compare ਹੈ । ਇਸ ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਦੋ ਸਟਿਰੰਗ ਮੁੱਲਾਂ ਚਿੰਨ੍ਹ /character (ASCII) ਦੇ ਨਾਲ ਤੁਲਨਾ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਅਤੇ ਤਿੰਨ ਮੁੱਲਾਂ ਨੂੰ ਵਾਪਿਸ ਕਰਦਾ ਹੈ ।

ਵਾਕ-ਰਚਨਾ (SYNTAX) :-

n=strcmp(string1,string2);

ਜਿੱਥੇ n ਇੱਕ ਇੰਟੀਜਰ ਵੇਰੀ◌ੇਏਬਲ ਹੈ।

| Value of n | Meaning |
|---|---|
| -1 | String1 < string2 |
| 0 | Both strings are same |
| 1 | String1>string2 |

Example (ਮਿਸਾਲ):-

n=strcmp("Good","GOOD");

# ਮਿਸਾਲ :- ਦੋ ਸਟਰਿੰਗਜ਼ ਦੀ ਤੁਲਨਾ ਕਰਨ ਲਈ ਪ੍ਰੋਗਰਾਮ

```
#include<stdio.h>
#include<string.h>
void main( )
{
int a;
a=strcmp("GOOD","Good");
If(a==0)
printf("Strings are equal");
else
printf("Strings are not equal");
getch( );
}
```

# strupr( )

ਫੰਕਸ਼ਨ strupr ਦਾ ਪੂਰਾ ਨਾਂ string upper ਹੈ। ਇਸ ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਦਿੱਤੇ ਸਟਰਿੰਗ ਮੁੱਲ ਨੂੰ ਅੰਗਰੇਜ਼ੀ ਦੇ ਵੱਡੇ ਅੱਖਰਾਂ ਵਿੱਚ ਤਬਦੀਲ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ।

ਵਾਕ-ਰਚਨਾ (SYNTAX) :-

strupr("string");

Example (ਮਿਸਾਲ ):-

strupr("school");

ਪਰਿਣਾਮ (Output):-

SCHOOL

# ਮਿਸਾਲ :- ਛੋਟੇ ਅੱਖਰਾਂ ਨੂੰ ਵੱਡੇ ਅੱਖਰਾਂ ਵਿੱਚ ਬਦਲਣ ਲਈ ਪ੍ਰੋਗਰਾਮ ।

```
#include<stdio.h>
#include<string.h>
void main( )
{
printf(strupr("school"));
getch( );
}
```

ਪ੍ਰੋਗਰਾਮ ਦਾ ਪਰਿਣਾਮ (Output of the Program):-

SCHOOL

# strlwr( )

ਫੰਕਸ਼ਨ strlwr ਦਾ  ਪੂਰਾ ਨਾਂ string lower ਹੈ। ਇਸ ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਦਿੱਤੇ ਸਟਰਿੰਗ ਮੁੱਲ ਨੂੰ ਅੰਗਰੇਜ਼ੀ ਦੇ ਛੋਟੇ ਅੱਖਰਾਂ ਵਿੱਚ ਤਬਦੀਲ ਕਰਨ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ।

ਵਾਕ-ਰਚਨਾ  (SYNTAX) :-

Strlwr(“string”);

ਮਿਸਾਲ  (Example):-

Strlwr(“SCHOOL”);

ਪਰਿਣਾਮ (Output) :-

school

# ਮਿਸਾਲ :- ਸਟਰਿੰਗ ਦੇ ਅੰਗਰੇਜ਼ੀ ਦੇ ਵੱਡੇ ਅੱਖਰਾਂ ਨੂੰ ਛੋਟੇ ਅੱਖਰਾਂ ਵਿੱਚ ਬਦਲਣ ਲਈ ਪ੍ਰੋਗਰਾਮ ।

```
#include<stdio.h>
#include<string.h>
void main( )
{
printf(strlwr("SCHOOL"));
getch( );
}
```

ਪ੍ਰੋਗਰਾਮ ਦਾ ਪਰਿਣਾਮ (Output of the Program) :-

school

# strrev( )

ਫੰਕਸ਼ਨ strrev ਦਾ ਪੂਰਾ ਨਾਂ string reverse ਹੈ। ਇਸ ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਦਿੱਤੇ ਗਏ ਸਟਰਿੰਗ ਨੂੰ ਉਲਟਾਉਣ ਲਈ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ।

ਵਾਕ-ਰਚਨਾ (SYNTAX) :-

strrev(“string”);

ਮਿਸਾਲ (Example) :-

strrev(“SCHOOL”);

ਪਰਿਣਾਮ (Output):-

LOOHCS

# ਮਿਸਾਲ :- ਸਟਰਿੰਗ ਨੂੰ ਉਲਟਾਉਣ ਲਈ ਪ੍ਰੋਗਰਾਮ ।

```
#include<stdio.h>
#include<string.h>
void main( )
{
printf(strrev("SCHOOL"));
getch( );
}
```

ਪ੍ਰੋਗਰਾਮ ਦਾ ਪਰਿਣਾਮ (Ouput of the Program ):-

LOOHCS

# ਯਾਦ ਰੱਖਣ ਯੋਗ ਗੱਲਾਂ

1. ਹਰ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ main() ਫੰਕਸ਼ਨ ਦਾ ਹੋਣਾ ਜਰੂਰੀ ਹੈ ।
2. ਕਿਸੇ ਵੀ ਸਟਰਿੰਗ ਫੰਕਸ਼ਨ ਦੀ ਵਰਤੋਂ ਕਰਨ ਲਈ ਹੈਡਰ ਫਾਈਲ ਨੂੰ ਪ੍ਰੋਗਰਾਮ ਵਿੱਚ ਸ਼ਾਮਿਲ ਕਰਨਾ ਪੈਂਦਾ ਹੈ ।
3. ਹੈਡਰ ਫਾਈਲ ਦਾ ਨਾਂ <string.h> ਹੈ ।
4. ਖਾਲੀ ਸਟਰਿੰਗ ਦੀ ਲੰਬਾਈ ਜੀਰੋ ਹੁੰਦੀ ਹੈ ।
5. ਹਰ ਸਟਰਿੰਗ ਦੀ ਸਮਾਪਤੀ ਖਾਲੀ ਚਿੰਨ੍ਹ ਨਾਲ ਹੋਣੀ ਚਾਹੀਦੀ ਹੈ ।