* Strings

→ String is a collection of characters or groups of sequence of characters.

→ The group of characters are stored in a "character array" is called a string.

→ Declaration

Syntax: datatype Stringname[size];

eg: char a[10];

eg: char a[6];

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]
R	A	M	U	\0	

Note: The compiler automatically assigns a Null char at the end of the string (\0)

Initialization: there are 2 types.

→ during declaration (compile time).

→ Runtime initialization

→ ① during declaration or compile time

(i) using single character constant

eg:

```
char a[6]={'R','A','M','U'};
```

(ii) using string constants.

eg: chas a[6]={"RAMU"}; (or) "RAMU";

Syntax: datatype stringname[size]={set of characters};

datatype stringname[size]={"total strings"};

→ Accessing %s

```
compile time
main()
{ char a[20]="vegetables";
  clrscr();
  printf("the given string is %s",a);
} getch();
```

Run time:

eg:
```
main()
{
char a[20];
clrscr();
printf("\n enter a string");
scanf("%s", a);
printf("\n the given string is %s", a);
getch();
}
```

→ There is no need of ampersand because string name itself specifies the base address of the string

Result:

input: enter a string

(i) Sai (ii) Sai Raj

output:

the given string is Sai the given string is Sai

Note: . scanf function does not consider the space as a character

→ to overcome this problem gets() is used

it is a function used to read a line of text even white spaces are occured in the string.

Syntax: gets(string name);

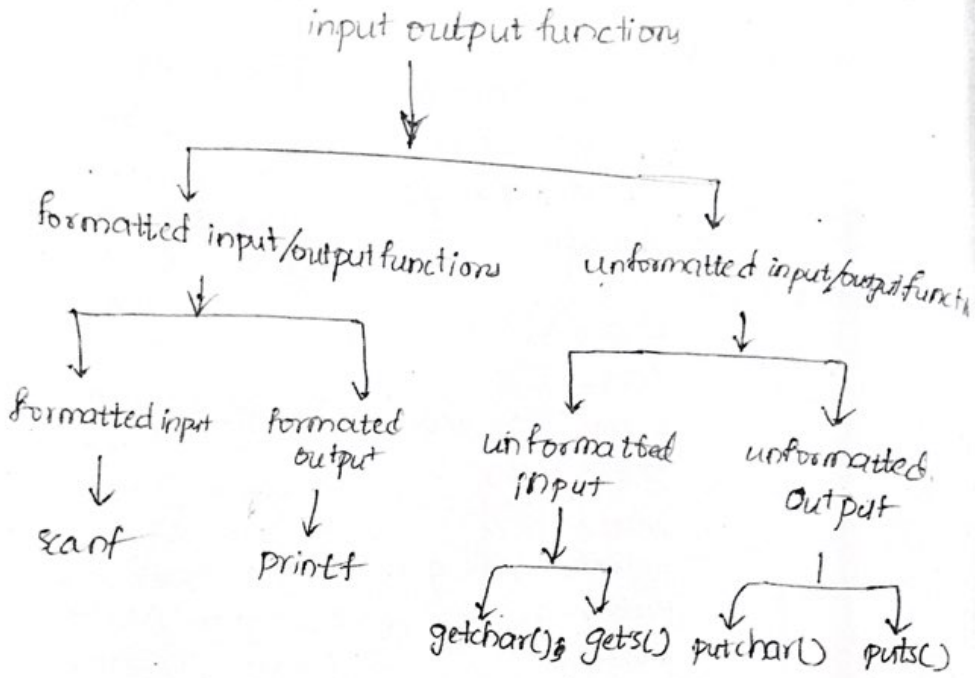
→ puts() used is to print a line of text

Syntax: puts(string name);

eg:
```
main()
{
char a[20];
clrscr();
printf("\n enter the string");
gets(a);
printf("\n the string is ");
puts(a);
getch();
}
```

input: enter the string
Sai Ram
output the string is
Sai Ram

* input output functions (with the help of



String handling functions:

* strlwr()
* strupr()
* strlen()
* strcpy()
* strncpy()
* strcat()
* strncat()
* strcmp()
* strncmp()
* strrev()
* strstr()

* strlwr(): used to convert the given string into lower case

syntax: strlwr(string name);

program:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
   char a[30];
   clrscr();
   printf("\n enter the string=\n");
   gets(a);
   strlwr(a);
   printf("\n the given string in lower case =");
   puts(a);
   getch();
}
```

input:
enter the string
SADHVIKA
output
the given string in lower case is
sadhvika

* strupr(): used to convert the given string into upper case

syntax: strupr(string name);

program:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
   char a[20];
   clrscr();
   printf("\n enter the string=\n");
   gets(a);
   strupr(a);
   printf("\n the given string in upper case =");
   puts(a);
   getch();
}
```

input
enter the string
sadhvika
output
the given in upper case is
SADHVIKA

3) strlen() - This function returns the length of the string i.e, the number of characters in a string.

Syntax:

integer variable = strlen(string name);

Program:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
   char a[30];
   int b;
   printf("enter the string=\n");
   gets(a);
   b=strlen(a);
   printf("length of string=%d",b);
   getch();
}
```

input
enter the string=hello
output
length of string=5

Note: NULL character is not considered as character

4) strcpy(): used to copy one string to another

Syntax: strcpy(destination string, source string)

eg:

1) chas a[50];
strcpy("Hello",a);
o/p: error

2) char a[50];
strcpy(a,"hello");
o/p: a="Hello"

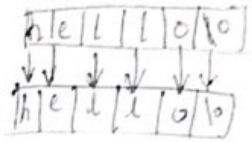
Note

The size of the destination string must be greater than or equal to ~~th~~ size of source string

$$DS \geq SS$$

program:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char a[50], b[50];
    clrscr();
    printf("enter a source string");
    gets(a);
    strcpy(b,a);
    printf("copied string=");
    puts(b);
    getch();
}
```



5) strncpy() :- used to copy the number of characters from source to destination

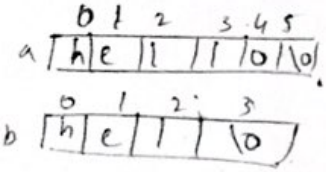
syntax:

```
strncpy(destination string, source string, n);
```

n → number of strings to be copied

program:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char a[50], b[50];
    clrscr();
    printf("enter a string");
    gets(a);
    strncpy(b,a,3);
    b[3]='\0';
    printf("copied string=");
    puts(b);
    getch();
}
```



6) strcat(): used to combine or concatenating two string

syntax:

```
strcat(string1, string2);
```

note: The length of resultant string must be greater than or equal to the sum of the length of two strings

Program:

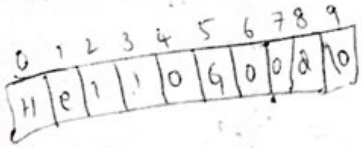
```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
 char a[50], b[20];
 clrscr();
 printf("enter the string A=");
 gets(a);
 printf("enter the string B=");
 gets(b);
 strcat(a,b);
 printf("concatenated string = %s", a);
 getch();
}
```

7) strncat(): used for combining the number of characters from string2 to ⟶ string1

syntax: strncat(string1, string2, n);

Program:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char a[30]="Hello";
    char b[20]="Good Morning";
    clrscr();
    strncat(a, b, 4);
    a[9]='\0';
    printf("concatenated string= %s", a);
} getch();
```



8) strcmp(): used to compare 2 strings

it returns the ASCII difference of the first two non-matching characters in both the strings

Syntax: strcmp(String1, String2);

if the difference is zero ⟹ string1 = string2

if the difference is +ve ⟹ string1 > string2

if the difference is -ve ⟹ string1 < string2

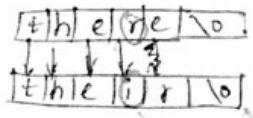
eg.

```
char a[10] = "there"
char b[10] = "their"
strcmp(a,b);
```

output: string1 > string2

A - Z ⇒ 65 - 90

a - z ⇒ 97 - 122



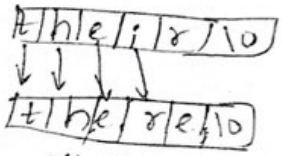
difference 'r' > 'i'

eg: char a[10] = "their"

char b[10] = "there"

strcmp(a,b);

output: string1 < string2



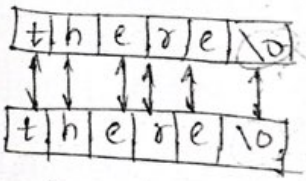
'i' < 'r'

eg: char a[10] = "there"

char b[10] = "there"

strcmp(a,b);

output: string1 = string2



program:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
  char a[50],b[50];
  int d;
  clrscr();
  printf("enter 2 strings");
```

```
    scanf("%s %s",a,b);
    d= strcmp(a,b);
    if(d==0)
      printf("%s is equal to %s",a,b);
    else
     if(d>0)
      printf("%s is greater than %s",a,b);
    else
     if(d<0)
      printf("%s is less than %s",a,b);
    getch();
}
```

9) strncmp(): This function is used for comparing first 'n' characters of 2 strings

Syntax:

```
strncmp(string1, string2, n);
```

eg
```
char a[10]="the";
char b[10]="there";
strncmp(a,b,3);
```

output: Both strings are equal

10) strrev(): the function is used for reversing a string
the reversed string will be stored in the same string

Syntax: strrev(string);

Program:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
   char a[50];
   clrscr();
```

```
printf("enter a string");
gets(a);
strrev(a);
printf(reversed string is = %s", a);
getch();
}
```

* getchar():-

It is a single character input function
It returns a single character
It does not require any argument

Syntax

```
character variable = getchar();
```

where the character variable refers to the name of the variable

* putchar():-

It is used to display single character on the output screen

Syntax:

```
putchar(char variable);
```

Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
  char b;
  clrscr();
  printf("\n enter a single character=");
  b=getchar();
  printf("the given character = ");
  putchar(b);
  getch();
}
```

Program:

The getchar() function can also be used to read multiple characters by reading one character at a time with the help of keyboard.

putchar() is also used to display multiple characters at a time by using for loop

```
# include <stdio.h>
# include <conio.h>
void main()
{
   char a[20];
   int n;
   printf("Enter the size of the string =\n");
   scanf("%d",&n);
   printf("enter the string=\n");
   for(i=0; i<n; i++)
   {
      a[i] = getchar();
   }
   a[i]='\0';
   printf("the given string=\n");
   for(i=0; i<n; i++)
   {
      putchar(a[i]);
   }
   getch();
}
```

* Character analysis and conversion functions

There are some predefined functions available in "ctype.h" library for analyzing the character input and converting them.

- Analysis functions

function	checks whether entered character is
1. isalpha()	An alphabet (or) not
2. isdigit()	A digit (or) not
3. isspace()	A space, a newline (or) tab
4. ispunct()	A special symbol or not
5. islower()	A lower case alphabet or not
6. isupper()	A upper case alphabet or not

- Converting functions

7. tolower()	Converts an upper case alphabet to lower case
8. toupper()	Converts an lowercase alphabet to upper case.

Program:

```
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
void main()
{
   char a='D';
   clrscr();
   if(isalpha(a))
      printf("%c is an alphabet",a);
   else
      printf("%c is not an alphabet",a);
   getch();
}
```

* String to number and number to string conversion

There →

1) sscanf(): used to converting string to number

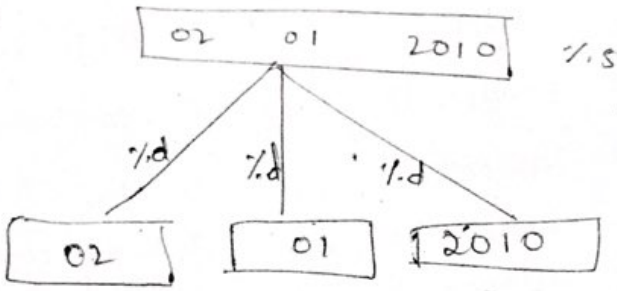
syntax:

sscanf(string name, "control string", &variable list);

eg: char a[2] = "17 11 2015";

sscanf(a, "%d %d %d", &date, &month, &year);

for eg,



Program:

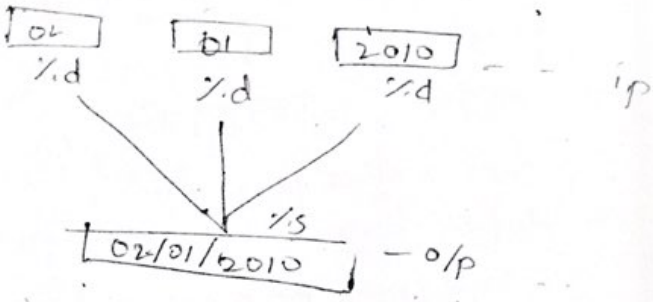
```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
   char a[20] = "02 01 2010";
   int day, mon, yr;
   clrscr();
   sscanf(a, "%d %d %d", &day, &mon, &yr);
   printf("Date = %d", day);
   printf("Month = %d", mon);
   printf("Year = %d", yr);
   getch();
}
```

3) Number to string conversion.

Sprintf() takes different numeric values as input & converts it into a single string.

· Syntax:

Sprintf (string name, "control string", variable list);



Program:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
  char a[50];
  int date=11, mon=12, yr=2014;
  clrscr();
  sprintf(a,"%d/%d/%d",date,mon,yr);
  printf("todays date = %s",a);
  getch();
}
```