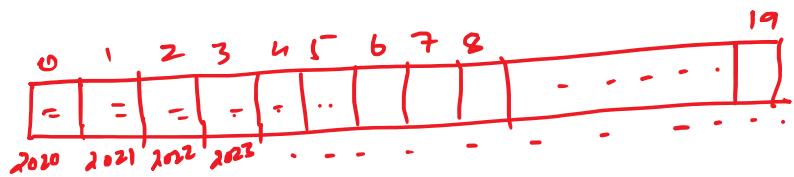


String in C →

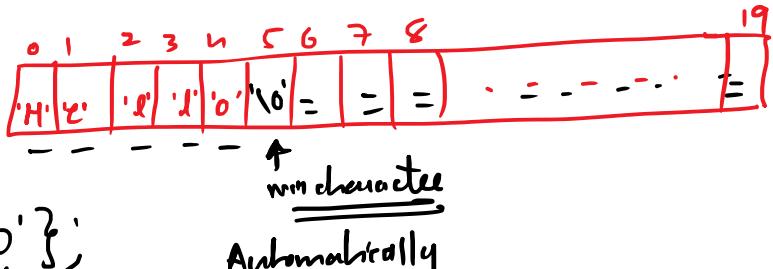
→ Array of characters.

Ex ✓ char s[20];



To Initialize →

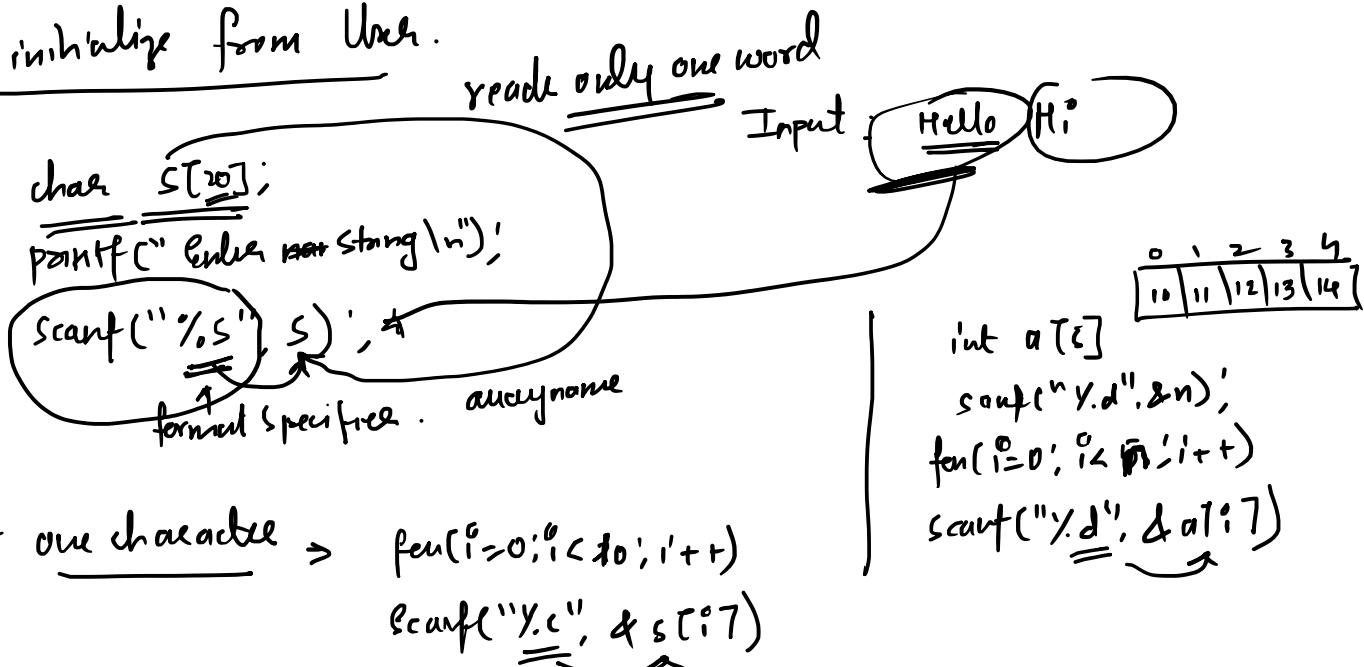
① char s[20] = "Hello";



② char s[20] = { 'H', 'e', 'l', 'l', 'o', 'D' };

Automatically  
(End of String)

To initialize from User.



\* one character →  $\text{for}(i=0; i<10; i++)$   
 $\text{scanf}("y.c", \&s[i])$

```
#include<stdio.h>
int main()
{
    char name[20];
    printf("enter name\n");
    scanf("%s", name);
```

↓  $\text{scanf}("y.c")$  read only one word

If input = sanjeev dwivedi  
will read and store only sanjeev  
in array

```

printf("enter name\n");
scanf("%s",name);
    =  

printf(" name = %s\n",name); → name = sanjeev
    ← in array
return 0;
}

```

To read a line use gets(arrayname):

```

#include<stdio.h>
int main()
{
    char name[20];
    printf("enter name\n");
    gets(name);
    printf(" name = %s\n",name);
    ← if input = sanjeev dwivedi
        ← it will read complete line and
        ← store in array.
    return 0;
}

```

To print String

char name[20] = "Hello"  
printf("%s", name); ⇒ Hello  
puts(name); ⇒ Hello

## (String.h) functions .

Return type:

① int strlen ( char [ ] )

arguments

⇒ gt will return length of the string passed to it as argument.

Ex      char st [20] = "Hello";      array size = 20  
int l = strlen (st);      string length = 5  
printf (" length = %d \n", l);      dp = 5

② int strcmp ( char [ ], char [ ] )      takes two strings as arg  
 returns 1 int value .

↳ gt returns the difference between two strings passed as argument.

↳ Diff is based on ASCII values

↳ gt is case Sensitive -

Ex)      char s1 [20] = "sanjay";  
char s2 [20] = "Sanjeev";

int d = strcmp (s1, s2); → perform s1 - s2

$$\begin{array}{r}
 s1 \quad \underline{s \ a \ n \ j \ a \ y} \\
 - \quad \underline{S \ a \ n \ j \ e \ e \ v} \\
 \hline
 \end{array}$$

→ 32      1..... Last NonZero value

$\rightarrow$  32  $\rightarrow$  Returns first NonZero value

$$d = 32 > 0 \quad \underline{s_1 > s_2}$$

Let  $s_1 = \begin{array}{ccccccccc} f & a & n & i & a & y \\ s & a & n & i & e & e & v \end{array}$   $\underline{\underline{10}} \quad \underline{\underline{s_1 < s_2}}$   
 $s_2 = \begin{array}{ccccccccc} s & a & n & i & e & e & v \\ o & o & o & o & -4 & = \end{array}$   
return  $\underline{-4}$

### ③ int strcmp( char [ ] , char [ ] )

↳ It returns diff betw two strings passed as argument.

↳ It is Case Insensitive

char s1[20] = "sanjay";  
char s2[20] = "Sanjeev";

int d = strcmp(s1, s2);

$\begin{array}{ccccccccc} s_1 & s & a & n & i & a & y & - \\ s_2 & S & a & n & i & e & e & v \\ \hline & o & o & o & o & -4 & = \end{array}$   
return  $\underline{\underline{-4}}$

### ④ void strcpy (char [ ] , char [ ] )

④ void strcpy (char [ ], char [ ] )

It copies the string passed as second arg to the string passed as first arg.

Ex      `char s1[20] = "Morning";`  
          `char s2[20] = "Hello";`  
          `strcpy(s1, s2);`  
          `puts(s1);`    → Hello  
          `puts(s2);`    → Hello.

⑤ void strcat (char [ ], char [ ] ),

It concatenates the string passed as second arg to the end of string passed as first argument.

Ex      `char s1[20] = "Morning";`  
          `char s2[20] = "Hello";`  
  
          `strcat(s1, s2);`  
          `puts(s1);` → MorningHello  
          `puts(s2);` → Hello

⑥ void strrev (char [ ] ) ⇒ It will reverse the ~~string~~ passed to it.

`char s1[20] = "Hello";` ✓  
      strrev(s1);  
                        ..-N

shrev(s1);  
putc(s1);     $\Rightarrow$  diff.

3) Write a Program to check a word is palindrome or not.

Assume String is  
palindrome.

~~f=1~~

i=0  
j = strlen(st)-1;

st

m	a	l	a	y	a	l	e	m	'0'	...
0	1	2	3	4	5	6	7	8	9	...

while (i < j)

if (st[i] != st[j])

{ f=0;

} break;

i++;

j--;

"if (f==1)  
print "Palindrome Mai";

else  
print "Palindrome Nahi";

```
#include<stdio.h>
#include<string.h>
int main()
{
    char s1[25];
    int i,j,f=1;

    printf("enter the string\n");
    gets(s1);

    i=0;
    j=strlen(s1)-1;
```

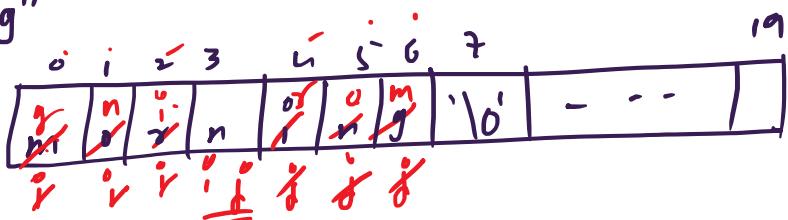
```
while(i<j)
{
    if(s1[i]!=s1[j])
    {
        f=0;
        break;
    }
    i++;
    j--;
}

if(f==0)
printf("string %s is not a palindrome\n",s1);
else
printf("string %s is a palindrome\n",s1);

return 0;
}
```

(13) WAP to reverse a String without using Built In function.

char s1[20] = "Morning";



$$i = 0;$$

$$f = \text{stolen}(\text{si})^{-1};$$

char t;

while ( $i < j$ )

$t = s1[i];$   
 $s1[i] = s1[j];$   
 $s1[j] = t;$ 
Swap character at pos<sup>n</sup>; & j<sup>o</sup>

`put(s1);`  $\Rightarrow$  g n i n r o m

```
#include<stdio.h>
#include<string.h>
int main()
{
int i,j;
char t;// for swapping characters
char s1[20];

printf("enter the string\n");
gets(s1);

printf("original string = %s\n",s1);
//reverse logic swap first , last then second and second last
```

```

i=0;
j=strlen(s1)-1;

while(i<j)
{   t=s1[i];
    s1[i]=s1[j];
    s1[j]=t;

    i++;
    j--;
}

} //while

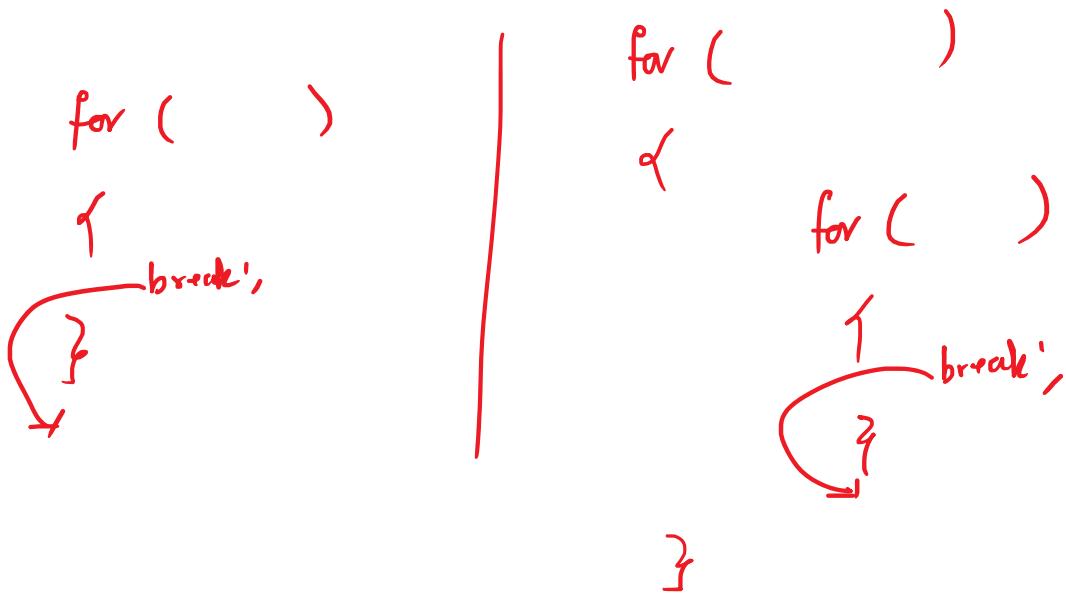
printf(" reverse string is %s \n",s1);

return 0;
}

```

Revise

break → terminates Current loop.



(continue) → terminates current iteration and goes to next iteration of current loop.

## Iteration of current loop

```
for( i=1; i<=5; i++)  
{   printf("Hello");  
    if( i>=3)  
        continue;  
    printf("ni");  
}
```

i=1 Hello H  
i=2 Hello H  
i=3 Hello }  
i=4 Hello }  
i=5 Hello }  
i=6

```
for ( ; i++)  
{  
    continue;  
}
```

```
for ( ; i++)  
{  
    for ( ; j++)  
    {  
        continue;  
    }  
}
```

Structure → It is User Defined Data Type.

→ It is Collection of diff data type.

typedef struct

```
    { char title[25]; }           25
    { char author[25]; }         25
    float price;                4
                                54 bytes (Size of Structure
```

Book;

Structure name

int x, y;

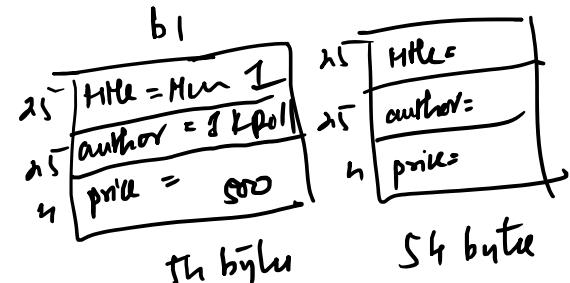
x & y are variable of type

int

x & y will have all properties of int

Created a Data Type →

To Create Variable, Book b1, b2;



```
printf("For First Book");
printf("Enter title\n");
gets(b1.title);
printf("Enter author\n");
gets(b1.author);
printf("Enter price\n");
scanf("%f", &b1.price);
```

```
printf("Enter details for Second Book\n");
printf("Enter title\n");
gets(b2.title);
printf("Enter Author\n");
gets(b2.author);
printf("Enter price\n");
scanf("%f", &b2.price);
```

Q1 WAP to read Title, Author, Price of 5 books using array of Structure . Display the records in Ascending order of price

```
#include<stdio.h>
typedef struct
{
    char title[20];
    char author[20];
    float price;
} Book;
```

```
int main()
```

```
{ Book b[5];
Book t;
int i, j;
```

```
printf("Enter details of Books\n");
```

```
for(i=0; i<5; i++)
```

```
{ printf("For Book %d\n", i+1);
```

fflush(stdin);

```
printf("Enter Title\n");
```

```
gets(b[i].title);
```

fflush(stdin);

```
printf("Enter Author\n");
```

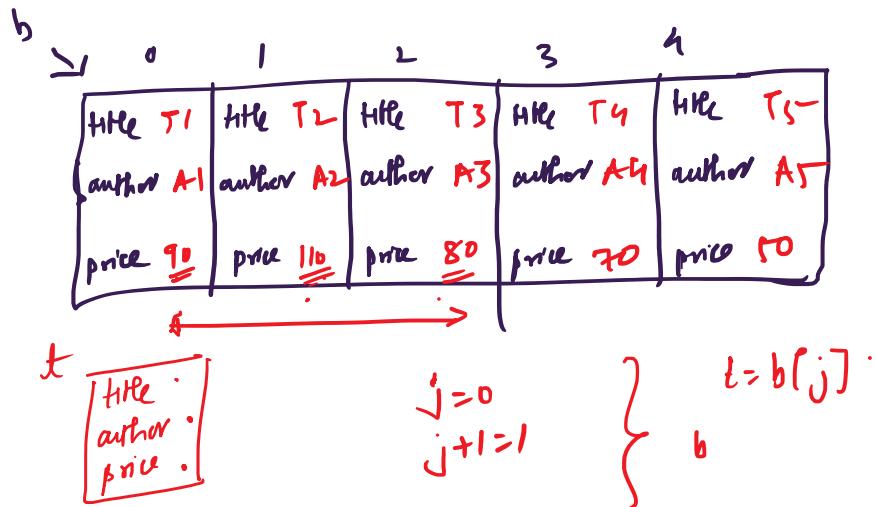
```
gets(b[i].author);
```

```
printf("Enter price\n");
```

```
scanf("%f", &b[i].price);
```

```
}
```

//Sort the Record.



```

for(i=0; i<4; i++)
{
    for(j=0; j<4; j++)
        if (b[j].price > b[j+1].price)
            {
                t = b[j];
                b[j] = b[j+1];
                b[j+1] = t;
            }
}

```

printf (" Details of Books in Ascending order of price\n");

```

for(i=0; i<5; i++)
{
    printf (" Title = %s\n", b[i].title);
    printf (" Author = %s\n", b[i].author);
    printf (" Price = %f\n", b[i].price);
}
return 0;
}

```

(11) #include <stdio.h>

typedef struct

```

    char playername[25];
    char teamname[25];
    float bat-avg;
} Player;

```

int main()

```

    Player p[10];
    int i;
    printf("Enter details of players\n");
    for(i=0; i<10; i++)
    {
        printf("For Player %d\n", i+1);
        printf("Enter playername\n");
        fflush(stdin);
        gets(p[i].playername);
        printf("Enter Team name\n");
        fflush(stdin);
        gets(p[i].teamname);
        printf("Enter Batting Average\n");
        scanf("%f", &p[i].bat-avg);
    }
}

```

0	1	2	3	4	5	6	7	8	9
name									
team									
avg									

{}

```
printf("Details of Players are \n");
for(i=0; i<10; i++)
{
    printf("Player name = %s\n", p[i].playername);
    printf("Team Name = %s\n", p[i].teamname);
    printf("Average = %.f\n", p[i].bat_avg);
}
return 0;
}
```

(14) #include <stdio.h>

typedef struct

    char name[20];

    int roll;

    float fee;

} Student;

int main()

    Student s[10];

    int i;

    printf("Enter details of students\n");

    for(i=0; i<10; i++)

        →     printf("For student %d\n", i+1);

        printf("Enter name\n");

        fflush(stdin);

        gets(s[i].name);

```
get(s[i].name);
printf("Enter Roll\n");
scanf("%d", &s[i].roll);
printf("Enter fees\n");
scanf("%f", &s[i].fees);
```

}

```
printf("Details of Student are \n");
for(i=0; i<10; i++)
```

```
{   printf("Name = %s\n", s[i].name);
    printf("Roll = %d\n", s[i].roll);
    printf("Fees = %f\n", s[i].fees);
```

}

```
return 0;
```

}

⑯ #include <stdio.h>

```
typedef struct
{
    char book_name[20];
    float price;
    int page_number;
    char author_name[20];
}
```

Book;

```
int main()
```

```
{    Book b[10];
```

```
int i;
```

```
printf("Enter details of Books\n");
```

```
for(i=0; i<10; i++)
```

```
{    printf("For Book %d\n", i+1);
    fflush(stdin);
    printf("Enter name\n");
    gets(b[i].book_name);
}
```

```
printf("Enter price\n");
```

```
scanf("%f", &b[i].price);
```

```

printf("Enter page no\n");
scanf("%d", &b[i].page-no);
fflush(stdin);
printf("Enter author name\n");
gets(b[i].author-name);
}

printf("Details of Book\n");
for(i=0; i<10; i++)
{
    printf("Book Name = %s\n", b[i].book-name);
    printf("Book Price = %f\n", b[i].price);
    printf("Book Page No = %d\n", b[i].page-no);
    printf("Book Author = %s\n", b[i].book-author);
}
return 0;
}

```

```
#include<stdio.h>
```

```
typedef struct
{
    char name[20];
    int roll;
    float per;
}Student;
```

```
int main()
{
```

```
    Student s[5];
```

```
    int i,p;
    float max;
    printf("enter details of students\n");
```

```
    for(i=0;i<5;i++)
    {
        printf("for %d student\n",i+1);
        printf("enter name\n");
        gets(s[i].name);
        printf("enter the roll\n");
        scanf("%d",&s[i].roll);
        printf("enter per\n");
        scanf("%f",&s[i].per);
        fflush(stdin);
    }
```

```
    max=s[0].per;
    p=0;
```

```
    for(i=1;i<5;i++)
    {
        if(s[i].per>max)
        {
            max=s[i].per;
            p=i;
        }
    }
```

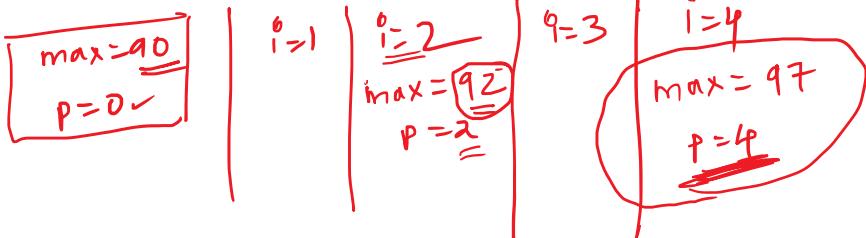
$\max = s[0].per$   
 $p = 0$

0	1	2	3	4
name A roll 11 per 90	name B roll 12 per 60	name C roll 13 per 92	name D roll 14 per 85	name E roll 15 per 97

Accept details for 5 Student & display details of student with highest percentage.

Read Data

0	1	2	3	4
name A roll 11 per 90	name B roll 12 per 60	name C roll 13 per 92	name D roll 14 per 85	name E roll 15 per 97



```
printf("details of student with max per\n");
printf("name= %s\n",s[p].name);
printf("roll= %d\n",s[p].roll);
printf("per= %f\n",s[p].per);
return 0;
}
```

#include <stdio.h>

void swap( int <sup>2020</sup> \*x, <sup>3030</sup> int \*y )

{  
    int t;

    t = \*x;  
    \*x = \*y;  
    \*y = t;  
}

int main()

{  
    int x=10, y=20;

    printf("Before Function call\n");

    printf("x=%d and y=%d\n", x,y);

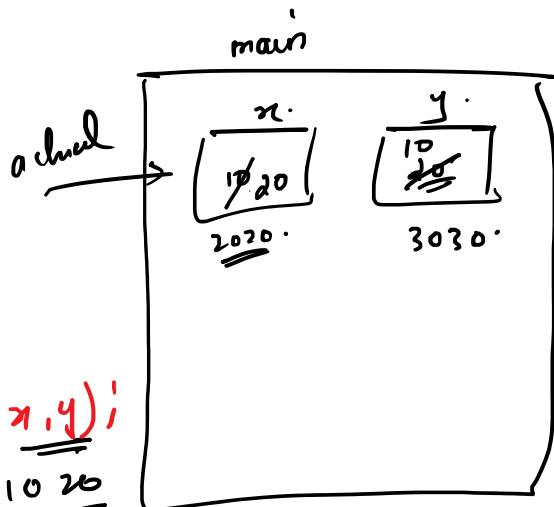
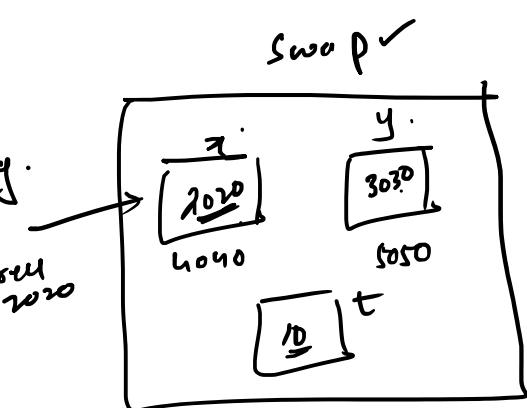
    swap( <sup>2020</sup> &x, <sup>3030</sup> &y );

    actual arg.

    printf("After Function call\n");

    printf("x=%d and y=%d\n", x,y);

}  
    return 0;



In call By Reference/Address:

The actual arg are address and formal arg are reference/pointer

Here actual & formal arg refers to same memory  
Hence changes made in formal arg are reflected in  
Actual arg.