

strings and pointers

Accessing string elements

1.

```
main()
{
    char str[] = "virat";
    char *ptr;
    ptr = str;
    for(i=0; *ptr !='\0';i++)
        printf("%c", *ptr++);
}
```

2.

```
main()
{
    char str[] = "virat";
    char *ptr;
    ptr = str;
    for(i=0;*(ptr+i)!='\0';i++)
        printf("%c", *(ptr+i));
}
```

3.

```
main()
{
    char str[] = "Virat";
    char *ptr = str;
    printf("string = %s", ptr);
}
```

4.

```
int main()
{
    char str[50];
    char *ptr = str;
    printf("Enter any string");
    gets(str); // welcome to cadmaxx
    printf("You have entered: %s", str);
}
```

1. count the num of chars in a string
2. convert a string in uc into lc
3. convert a string in lc into uc
4. check if a given char is a number
5. count the number of words in a string
6. check if a given char exist in the array
7. count the number of digits in a string

8. reverse the given string
9. copy one string into other
10. string concatenation(merge two strings)
11. compare two strings
  - if str1 = str2 = print- 0
  - str1 > str2 = print +ve value
  - str1 < str2 = -ve value
12. find a sub string in a given string
  - str1[] = "welcome to Cadmaxx";
  - str2[] = "come";
13. check if a sub string exist in a string matching whole chars
  - str1[] = "please come to cadmaxx"
  - str2[]="come"
14. replace the occurance of space with string "xyz"
  - str1[] = "welcome to cadmaxx"
  - str1[] = "welcomexyztoxyzcadmaxx"
15. WAP to comapre strings in reverse order
  - str1[] = "welcome ot cadmaxx"
  - str2[]="to"

```
main()
{
    char name[50];
    char *ptr = str;
    int nc=0;
    gets(str); //virat kohli
    while(*ptr !='\0')
    {
        nc++;
        ptr++;
    }
    printf("No of chars = %d", nc);
}
```

```
main()
{
    char name[50];
    char *ptr = str;
    int i;
    int nc=0;
    gets(str); //virat kohli
    while(*(ptr+i)!='\0')
    {
```

```

        nc++;
        i++;
    }
    printf("No of chars = %d", nc);
}

```

string pointer

```

main()
{
    char *ptr = "hello";
    printf("string = %s", ptr);
}

```

```

main()
{
    char *ptr = "Hello";
    int nc;
    while(*ptr!='\0')
    {
        nc++;
        ptr++;
    }
    printf("Nc = %d", nc);
}

```

```

main()
{
    char str[] = "hello";
    char *ptr = str;
    str[0] = 'f';
    *ptr='m';
    printf("%s", ptr);
}

```

```

main()
{

    char *ptr = "hello"
    *ptr='m';
    printf("%s", ptr);
}

```

strings and functions

strings, arrays, pointers, functions, call by value, call by reference

Printing string chars using functions

```
main()
{
    char a[]="Virat";
    for(i=0;a[i]!='\0';i++)
        display(a[i]);
}
```

```
void display(char *ptr)
{
    printf("%c", *ptr);
}
```

Note: The above program calls function for each char print separately, hence not recommended

```
main()
{
    int a[]={1,2,3,4,5};
    display(a);
}
```

```
void display(int *ptr)
{
    for(i=0;i<5;i++)
        printf("%d", *(ptr+i));
}
```

```
main()
{
    char a[]="Virat";
    display(a);
}
```

```
void display(char *ptr)
{
    for(i=0;i<5;i++)
        printf("%c", *(ptr+i));
}
```

count the number of chars in a string

```
main()
{
    char str[50];
}
```

```
int nc;
printf("Enter the string: ");
gets(str);
nc = char_count(str);
//nc = strlen(str);
printf("Num of chars = %d", nc);
}
```

```
int char_count(char *ptr)
{
    int n=0;
    while(*ptr!='\0')
    {
        n++;
        ptr++;
    }
    return n;
}
```

```
builin function n =strlen(str);
```