1. Write a program to find the length of a string without using `strlen()`.

IPO
INPUT: A string (`str`)
PROCESS: Count characters
OUTPUT: Length of the string

CODE;

```
#include <stdio.h>

void main()

{

    char str[100];

    int i = 0;

    scanf("%s", str);

    while(str[i] != '\0')

        i++;

    printf("Length of string = %d", i);

}
```

OUTPUT;

Output

```
hello
Length of string = 5
```

2. Copy one string to another

IPO

INPUT: A string (`str1`)

PROCESS: Copy characters to `str2`

OUTPUT: Copied string in `str2`

CODE;

```
#include <stdio.h>

void main()

{

    char str1[100], str2[100];

    int i;

    scanf("%s", str1);

    for(i = 0; str1[i] != '\0'; i++)

        str2[i] = str1[i];

    printf("Copied string: %s", str2);

}
```

OUTPUT;

```
Output

hello
Copied string: hello
```

3. Concatenate two strings

IPO
 INPUT: Two strings (`str1`, `str2`)
 PROCESS: ADD `str2` to `str1`
 OUTPUT: Concatenated string

CODE;

```
#include <stdio.h>

void main()

{

    char str1[100], str2[100];

    int i = 0, j = 0;

    scanf("%s", str1);

    scanf("%s", str2);

    while(str1[i] != '\0')

        i++;

    while(str2[j] != '\0')

    {

        str1[i] = str2[j];

        i++;

        j++;

    }

    printf("Concatenated string: %s", str1);

}
```

OUTPUT;

```
Output
welcome home
Concatenated string: welcomehome
```

4. Compare two strings

IPO
 INPUT: Two strings
 PROCESS: Compare each character
 OUTPUT: Equal or not

CODE;

```
#include <stdio.h>

void main()

{

    char s1[100], s2[100];

    int i, flag = 0;

    scanf("%s", s1);

    scanf("%s", s2);

    for(i = 0; s1[i] != '\0' || s2[i] != '\0'; i++)

    {

        if(s1[i] != s2[i])

        {

            flag = 1;

            break;
```

```
        }
    }
    if(flag == 0)
        printf("Strings are equal");
    else
        printf("Strings are not equal");
}
```

OUTPUT;

```
Output
welcome welcome
Strings are equal
```

5. Count vowels and consonants

IPO
INPUT: A string
PROCESS: Check each letter
OUTPUT: Number of vowels and consonants

CODE;

```
#include <stdio.h>
void main()
{
    char str[100];
    int i, vowels = 0, consonants = 0;
    scanf("%s", str);
```

```
    for(i = 0; str[i] != '\0'; i++)
    {
       if(str[i]=='a'||str[i]=='e'||str[i]=='i'||
         str[i]=='o'||str[i]=='u'||str[i]=='A'||
         str[i]=='E'||str[i]=='I'||str[i]=='O'||str[i]=='U')
            vowels++;
        else
            consonants++;
    }
    printf("Vowels = %d\n", vowels);
    printf("Consonants = %d", consonants);
}
```

OUTPUT;

Output

```
saveetha
Vowels = 4
Consonants = 4
```

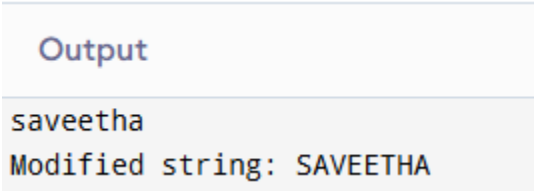
6. Convert lowercase to uppercase and vice versa

IPO
 INPUT: A string
 PROCESS: Toggle case of each letter
 OUTPUT: Modified string

CODE;

```
#include <stdio.h>

void main()

{

    char str[100];

    int i;

    scanf("%s", str);

    for(i = 0; str[i] != '\0'; i++)

    {

        if(str[i] >= 'a' && str[i] <= 'z')

            str[i] -= 32;

        else if(str[i] >= 'A' && str[i] <= 'Z')

            str[i] += 32;

    }

    printf("Modified string: %s", str);

}
```

OUTPUT;




7. Check if a string is a palindrome

IPO
 INPUT: A string
 PROCESS: Reverse and compare
 OUTPUT: Palindrome or not

CODE;

```
#include <stdio.h>

void main()

{

    char str[100];

    int i, len = 0, flag = 0;

    scanf("%s", str);

    while(str[len] != '\0')

        len++;

    for(i = 0; i < len / 2; i++)

    {

        if(str[i] != str[len - i - 1])

        {

            flag = 1;

            break;
```

```
        }
    }
    if(flag == 0)
        printf("Palindrome");
    else
        printf("Not Palindrome");
}
```

OUTPUT;

```
Output
malayalam
Palindrome
```

## 8. Reverse a string

IPO
 INPUT: A string
 PROCESS: Swap front and back
 OUTPUT: Reversed string

CODE;

```
#include <stdio.h>
void main()
{
    char str[100];
    int i, len = 0, temp;
    scanf("%s", str);
    while(str[len] != '\0')
```

```
        len++;
    for(i = 0; i < len / 2; i++)
    {
        temp = str[i];
        str[i] = str[len - i - 1];
        str[len - i - 1] = temp;
    }
    printf("Reversed string: %s", str);
}
```

OUTPUT;

```
Output
welcome
Reversed string: emoclew
```

## 10. Frequency of each character

IPO
INPUT: A string
PROCESS: Count frequency using ASCII index
OUTPUT: Each character and count

CODE;

```
#include <stdio.h>
void main()
{
    int num, digit;
    int freq[10] = {0};
```

```
    scanf("%d", &num);
    while(num > 0)
    {
        digit = num % 10;
        freq[digit]++;
        num = num / 10;
    }
    for(int i = 0; i < 10; i++)
    {
        if(freq[i] > 0)
            printf("%d = %d\n", i, freq[i]);
    }
}
```

OUTPUT;

Output

```
1122331
1 = 3
2 = 2
3 = 2
```