Day : Strings (7-8-2025)

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

   IPO

   Input: A string from the user.

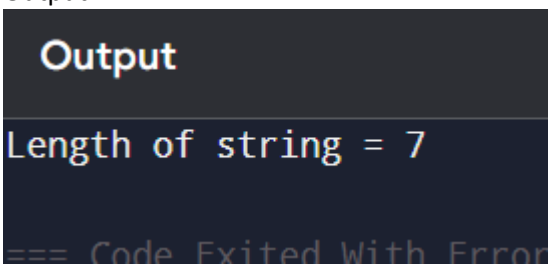
   Process: Read the string character by character until '\0' is reached, incrementing a counter for each character.

   Output: Length of the string.

   Program

```
#include <stdio.h>
void main()
{
   char str[] = "welcome";
   int i, length = 0;
   for (i = 0; str[i] != '\0'; i++)
   {
      length++;
   }
   printf("Length of string = %d", length);
}
```

   Output



2. Write a program to copy one string to another.

   IPO

   Input: A string from the user.

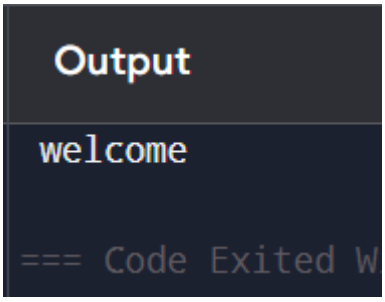
   Process: Read the string and copy each character into another string until '\0' is reached.

   Output: The copied string.

   Program

```
#include <stdio.h>
void main()
{
   char str1[] = "welcome";
   char str2[20];
   int i;
   for (i = 0; str1[i] != '\0'; i++)
   {
      str2[i] = str1[i];
   }
   str2[i] = '\0';
   printf(" %s", str2);
}
```

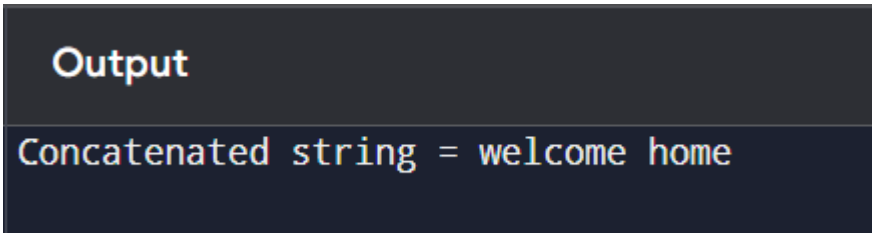
   Output



3. Write a program to concatenate two strings.
   IPO
   Input: Two strings from the user.
   Process: Append characters of the second string to the end of the first string until '\0' is reached.
   Output: The concatenated string.
   Program

```
#include <stdio.h>
#include <string.h>
void main()
{
   char str1[20] = "welcome ";
   char str2[] = "home";
   strcat(str1, str2);
   printf("Concatenated string = %s", str1);
}
```

   Output



4. Write a program to compare two strings.

   IPO

   Input: Two strings from the user.

   Process: Compare characters of both strings one by one until a mismatch is found or both strings end.

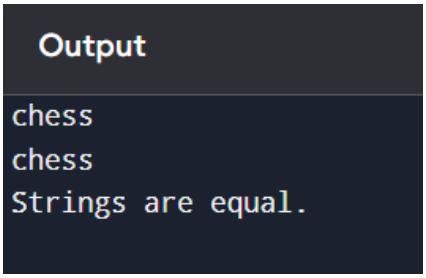
   Output: Message stating whether the strings are equal or different.

   Program

```
#include <stdio.h>


void main()
```

```
{
    char str1[100], str2[100];
    int i = 0, flag = 0;
    scanf("%s", str1);
    scanf("%s", str2);
    while (str1[i] != '\0' || str2[i] != '\0')
    {
        if (str1[i] != str2[i])
        {
            flag = 1; // strings are not equal
            break;
        }
        i++;
    }

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

Output



5. Write a program to count vowels and consonants in a string.
   IPO
   Input: A string from the user.

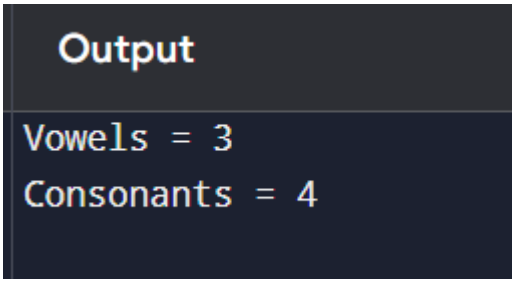
Process: Traverse the string; if the character is a vowel (a,e,i,o,u), increase vowel count, else if it's an alphabet letter, increase consonant count.

Output: Number of vowels and consonants.

Program

```
#include <stdio.h>
void main()
{
   char str[] = "welcome";
   int i, v = 0, c = 0;
   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')
         v++;
      else
         c++;
   }
   printf("Vowels = %d\nConsonants = %d", v, c);
}
```

output



6. Write a program to convert lowercase to uppercase and vice versa.

IPO

Input: A string from the user.

Process: Traverse the string and change each lowercase letter to uppercase, and each uppercase letter to lowercase.

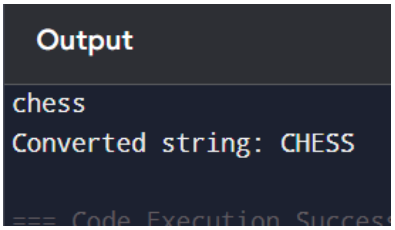
Output: The converted string.

Program

```
#include <stdio.h>

int main()

{

   char str[100];

   int i;

   scanf("%[^\n]", str);

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

```
{
      if (str[i] >= 'a' && str[i] <= 'z')
     {
         str[i] = str[i] - 32;
      }
      else if (str[i] >= 'A' && str[i] <= 'Z')
    {
         str[i] = str[i] + 32;
      }
  }
  printf("Converted string: %s", str);
  return 0;
}
```

Output

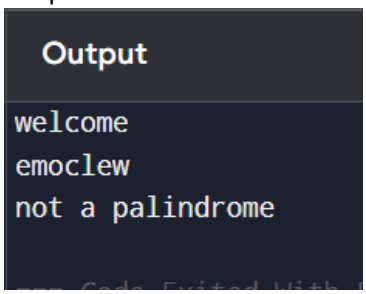


7. Write a program to check if a string is palindrome.
   IPO
   Input: A string from the user.
   Process: Compare characters from the beginning and end moving towards the middle; if all match, it's a palindrome.
   Output: Message stating whether the string is a palindrome.
   Program

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

void main()
{
  char str1[100] = "welcome";
  char str2[100];
  char str3[100];
  int i, j;
```

```
    for (i = 0; i < strlen(str1); i++)
        str2[i] = str1[i];
    str2[i] = '\0';

    printf("%s\n", str2);

    int l = 0;
    int flag = 0;
    for (i = strlen(str1) - 1, j = 0; i >= 0; i--, j++)
        str3[j] = str1[i];

    str3[j] = '\0';

    printf("%s\n", str3);

    if (strcmp(str2, str3) == 0)
        printf("palindrome");
    else
        printf("not a palindrome");
}
```

Output



8. Write a program to reverse a string.

   IPO

   Input: A string from the user.

   Process: Read the string and print characters in reverse order using indexing or by swapping.

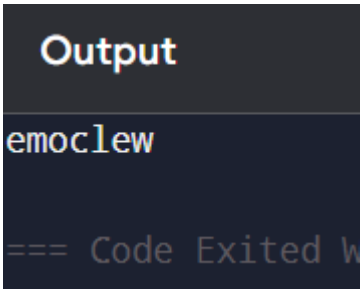
   Output: The reversed string.

   Program

```
#include <stdio.h>

#include <string.h>


void main()

{
```

```
    char str[] = "welcome";
    int len = strlen(str); // int instead of char
    int i;

    for (i = len - 1; i >= 0; i--)
    {
       printf("%c", str[i]);
    }
}
```

Output



9. Write a program to count words in a string.

IPO

Input: A string from the user.

Process: Traverse the string and count the number of spaces to determine word count

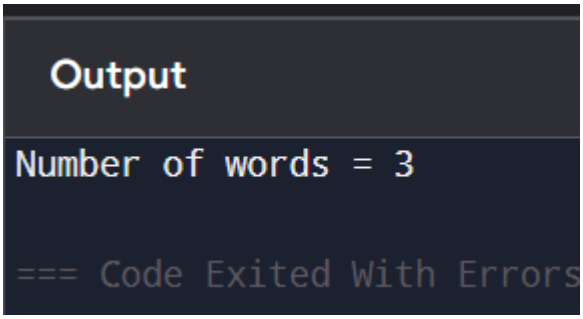
Output: Number of words in the string.

Program

```
#include <stdio.h>
void main()
  {
     char str[] = "welcome to home";
     int i, count = 1;
     for (i = 0; str[i] != '\0'; i++)
     {
```

```
        if (str[i] == ' ')
            count++;
    }
    printf("Number of words = %d", count);
}
```

Output



10. Write a program to find the frequency of each character in a string.

IPO

Input: A string from the user.

Process: For each character, count how many times it appears in the string

Output: Frequency of each character.

Program

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[100];
    int freq[256] = {0};
    int i;
    scanf("%[^\n]", str);
    for (i = 0; str[i] != '\0'; i++)
```

```
    {
        freq[(unsigned char)str[i]]++;
    }
    for (i = 0; i < 256; i++)
    {
        if (freq[i] != 0)
            printf("%c = %d\n", i, freq[i]);
    }

    return 0;
}
```

output