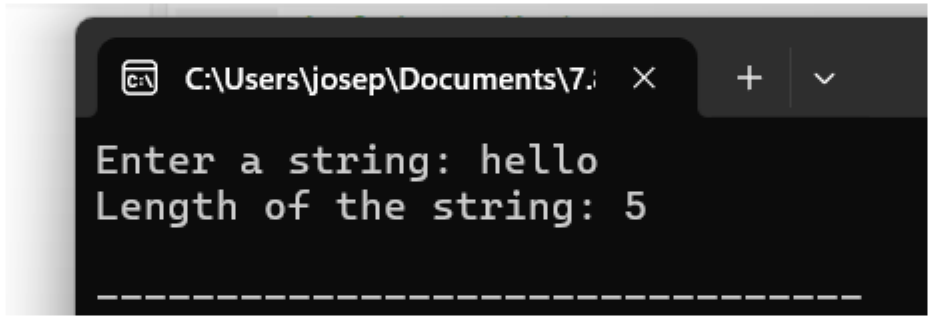
1. Write a program to find the length of a string without using strlen().

Input: A string entered by the user

Process: Count the number of characters manually using a loop until `'\0'`

Output: Length of the string

```
#include <stdio.h>

int main()
{
   char str[100];
   int length = 0;

   // Input
   printf("Enter a string: ");
   fgets(str, sizeof(str), stdin);

   // Process: Count characters until null character
   while (str[length] != '\0')
{
      if (str[length] == '\n') break; // ignore newline
      length++;
   }

   // Output
   printf("Length of the string: %d\n", length);

   return 0;
}
```


C:\Users\josep\Documents\7.

```
Enter a string: hello
Length of the string: 5

--------------------------------
```

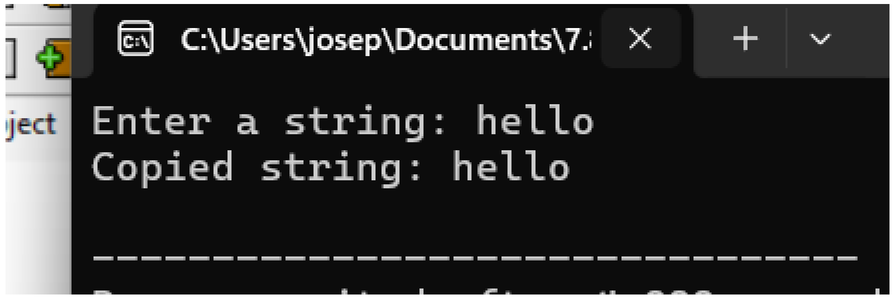
Write a program to copy one string to another.

Input: A string entered by the user

Process: Copy each character from the source string to the destination string manually using a loop

Output: The copied string

```
#include <stdio.h>
int main()
{
   char source[100], destination[100];
   int i = 0;
   printf("Enter a string: ");
   fgets(source, sizeof(source), stdin);
   while (source[i] != '\0') {
      destination[i] = source[i];
      i++;
   }
   destination[i] = '\0';
   printf("Copied string: %s", destination);
   return 0;
}
```



```
Enter a string: hello
Copied string: hello

--------------------------------
```

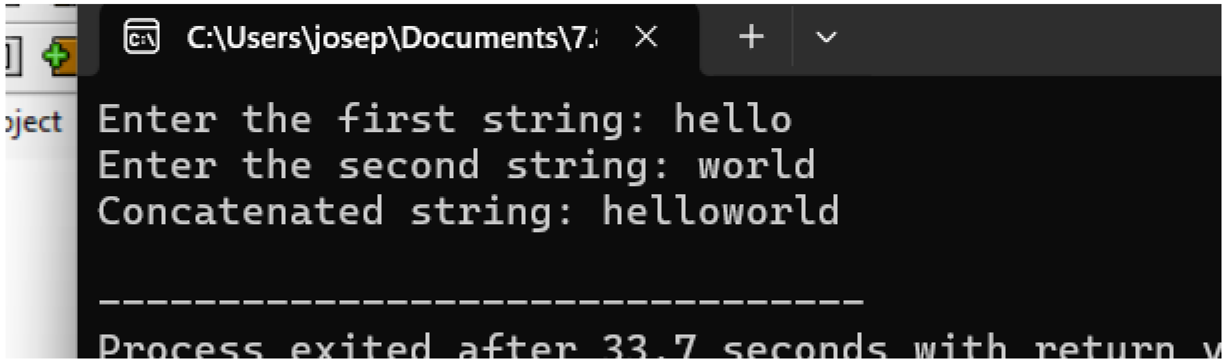
3. Write a program to concatenate two strings.

Input: Two strings

Process: Append second string to the end of the first manually

Output: Concatenated string

```
#include <stdio.h>
int main()
{
   char str1[100], str2[100];
   int i = 0, j = 0;
   printf("Enter first string: ");
   fgets(str1, sizeof(str1), stdin);

   printf("Enter second string: ");
   fgets(str2, sizeof(str2), stdin);
   while (str1[i] != '\0') {
      if (str1[i] == '\n') str1[i] = '\0';
      i++;
   }
   while (str2[j] != '\0')
{
      if (str2[j] == '\n') break;
      str1[i++] = str2[j++];
   }
   str1[i] = '\0';
   printf("Concatenated string: %s\n", str1);
   return 0;
}
```



```
Enter the first string: hello
Enter the second string: world
Concatenated string: helloworld

--------------------------------
Process exited after 33.7 seconds with return v
```

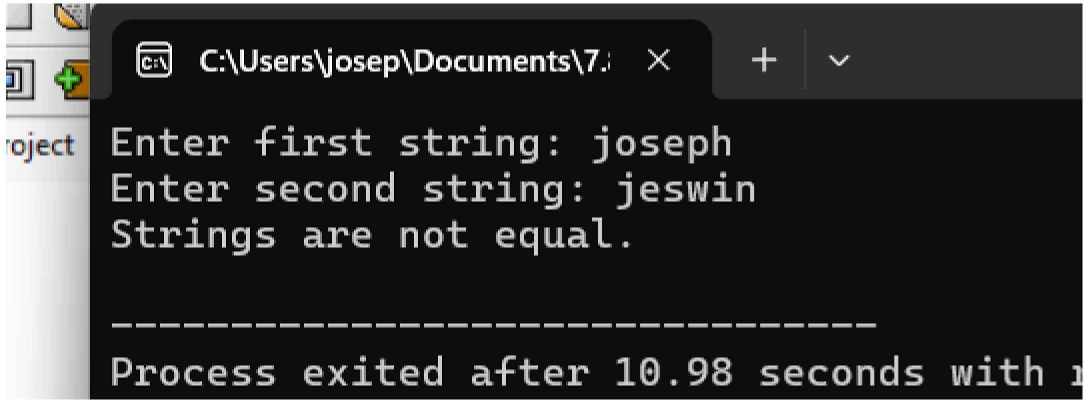
4. Write a program to compare two strings.

Input: Two strings

Process: Compare characters one by one from both strings

Output: Display whether strings are equal or not

```
#include <stdio.h>
int main()
{
   char str1[100], str2[100];
   int i = 0, flag = 0;
   printf("Enter first string: ");
   fgets(str1, sizeof(str1), stdin);
   printf("Enter second string: ");
   fgets(str2, sizeof(str2), stdin);
   while (str1[i] != '\0' || str2[i] != '\0')
{
      if (str1[i] != str2[i])
{
         flag = 1;
         break;
      }
      i++;
   }
   if (flag == 0)
      printf("Strings are equal.\n");
   else
      printf("Strings are not equal.\n");
   return 0;
}
```



```
Enter first string: joseph
Enter second string: jeswin
Strings are not equal.

--------------------------------
Process exited after 10.98 seconds with r
```

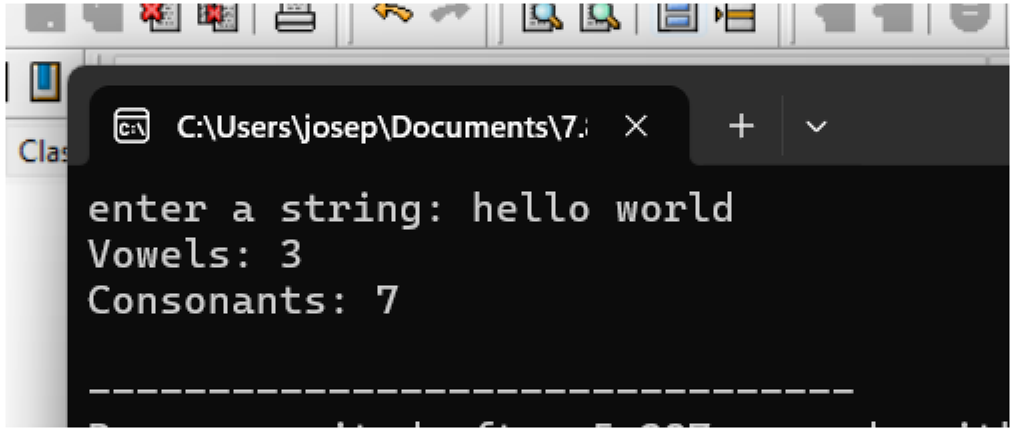
5. Write a program to count vowels and consonants in a string.

Input: A string entered by the user

Process: Check each character; if it is a vowel, increment vowel count, else if it is an alphabet, increment consonant count

Output: Number of vowels and consonants

```
#include <stdio.h>
#include <ctype.h>
int main()
{
  char str[100];
  int i = 0, vowels = 0, consonants = 0;
  printf("enter a string: ");
  fgets(str, sizeof(str), stdin);
  while (str[i] != '\0')
{
    char ch = tolower(str[i]);
    if (isalpha(ch))
{
      if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
        vowels++;
      else
        consonants++;
    }
    i++;
  }
  printf("Vowels: %d\n", vowels);
  printf("Consonants: %d\n", consonants);
  return 0;
}
```



```
enter a string: hello world
Vowels: 3
Consonants: 7

--------------------------------
```

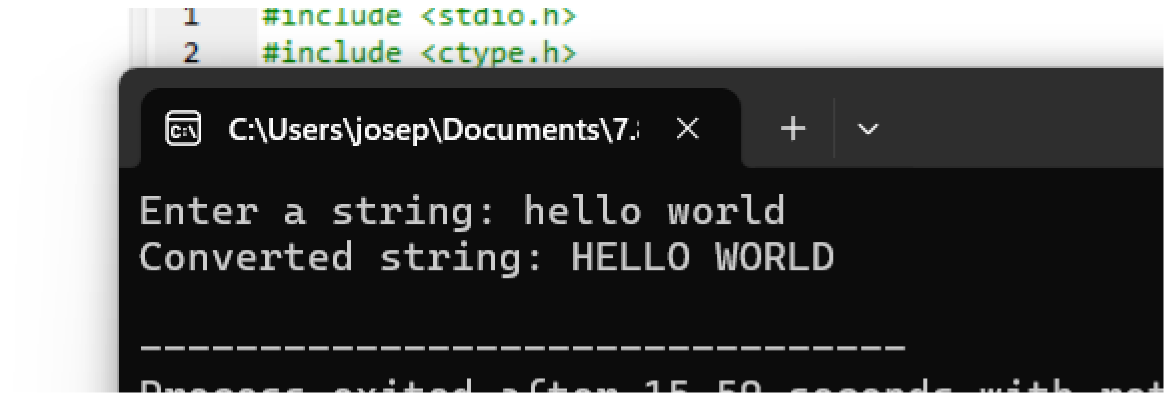
6. Write a program to convert lowercase to uppercase and vice versa.

Input: A string entered by the user

Process: Traverse each character; if lowercase, convert to uppercase and vice versa

Output: Display the converted string

```
#include <stdio.h>
#include <ctype.h>
int main()
{
   char str[100];
   int i = 0;
   printf("Enter a string: ");
   fgets(str, sizeof(str), stdin);
   while (str[i] != '\0') {
      if (islower(str[i]))
         str[i] = toupper(str[i]);
      else if (isupper(str[i]))
         str[i] = tolower(str[i]);
      i++;
   }
   printf("Converted string: %s", str);
   return 0;
}
```



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

```
Enter a string: hello world
Converted string: HELLO WORLD

--------------------------------
Process exited after 15.59 seconds with ret
```

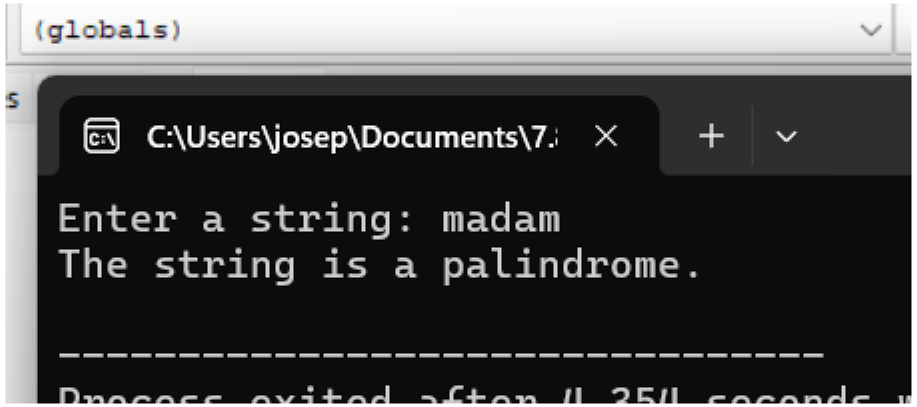
7. Write a program to check if a string is palindrome.

Input: A string entered by the user

Process: Compare characters from the start and end of the string moving toward the center

Output: Display whether the string is a palindrome or not

```
#include <stdio.h>
#include <string.h>
int main()
{
   char str[100];
   int i, len, flag = 0;
   printf("Enter a string: ");
   fgets(str, sizeof(str), stdin);
   len = strlen(str);
   if (str[len - 1] == '\n') {
      str[len - 1] = '\0';
      len--;
   }
   for (i = 0; i < len / 2; i++)
{
      if (str[i] != str[len - i - 1])
{
         flag = 1;
         break;
      }
   }
   if (flag == 0)
      printf("The string is a palindrome.\n");
   else
      printf("The string is not a palindrome.\n");

   return 0;
}
```



```
Enter a string: madam
The string is a palindrome.

--------------------------------
Process exited after 4.354 seconds w
```

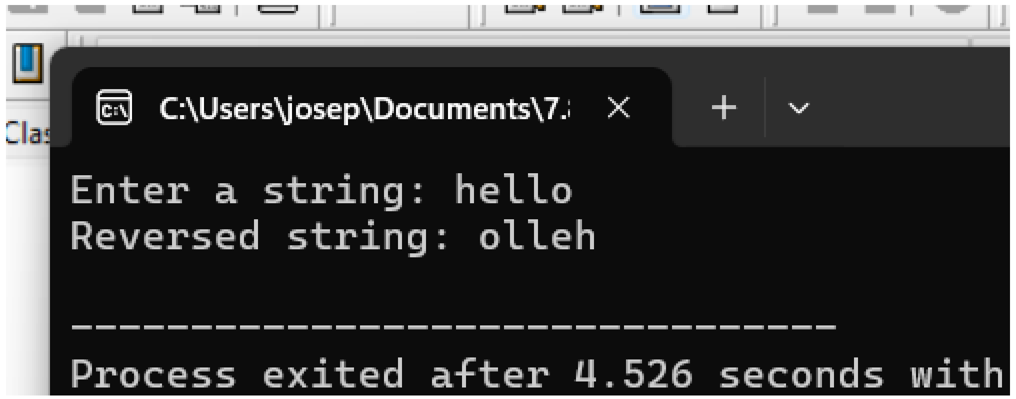
8. Write a program to reverse a string.

Input: A string entered by the user

Process: Swap characters from start and end of the string moving toward the center

Output: Display the reversed string

```
#include <stdio.h>
#include <string.h>
int main()
{
   char str[100];
   int i, len, temp;
   printf("Enter a string: ");
   fgets(str, sizeof(str), stdin);
   len = strlen(str);
   if (str[len - 1] == '\n')
{
      str[len - 1] = '\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\n", str);
   return 0;
}
```



```
Enter a string: hello
Reversed string: olleh

--------------------------------
Process exited after 4.526 seconds with
```

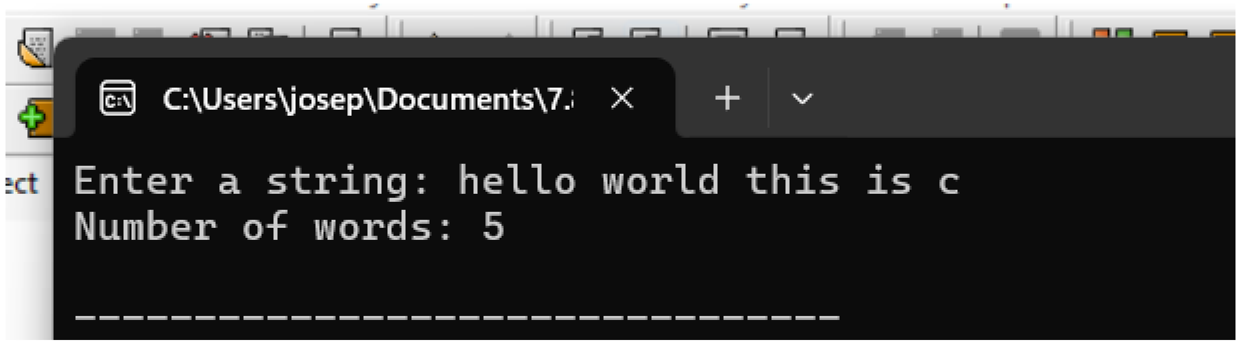
9. Write a program to count words in a string.

Input: A string entered by the user

Process: Count spaces between words, skipping multiple spaces

Output: Number of words in the string

```
#include <stdio.h>
#include <ctype.h>
int main()
{
   char str[200];
   int i = 0, word_count = 0;
   printf("Enter a string: ");
   fgets(str, sizeof(str), stdin);
   while (str[i] != '\0') {
      if ((i == 0 && !isspace(str[i])) ||
         (isspace(str[i - 1]) && !isspace(str[i]) && str[i] != '\n')) {
         word_count++;
      }
      i++;
   }
   printf("Number of words: %d\n", word_count);
   return 0;
}
```



```
Enter a string: hello world this is c
Number of words: 5

--------------------------------
```

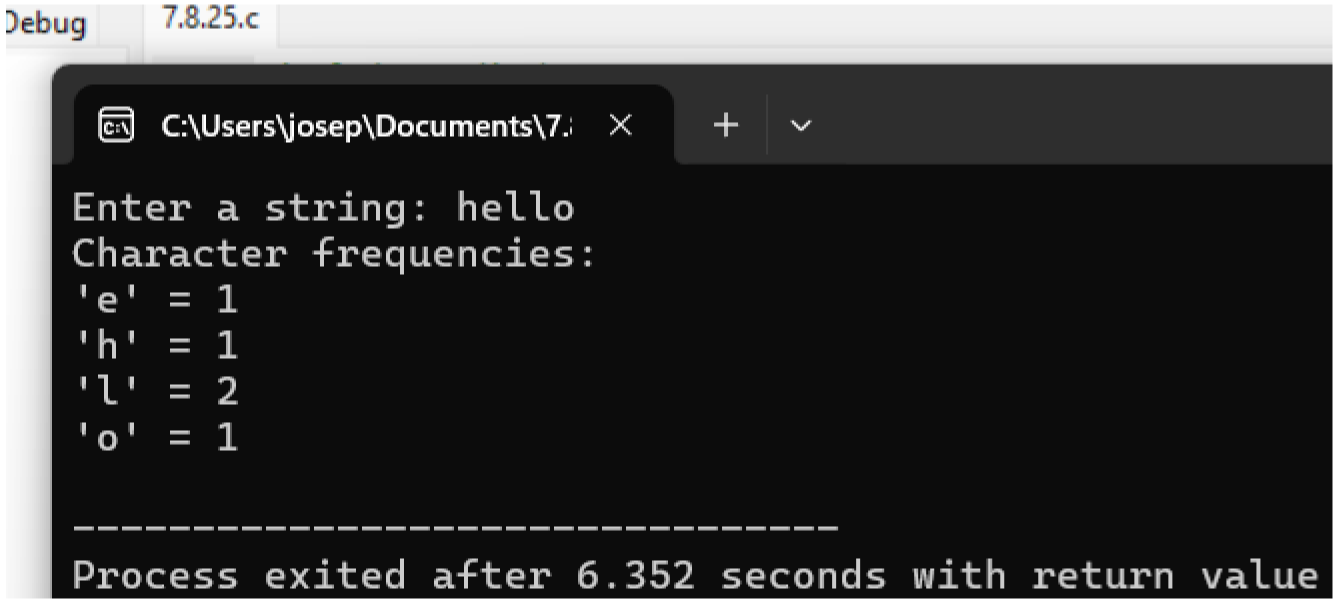
```c
#include <stdio.h>
#include <string.h>

int main() {
    char str[200];
    int freq[256] = {0}; // Frequency array for all ASCII characters
    int i;

    // Input
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    // Process: Count frequency
    for (i = 0; str[i] != '\0'; i++) {
        if (str[i] != '\n') {
            freq[(unsigned char)str[i]]++;
        }
    }

    // Output
    printf("Character frequencies:\n");
    for (i = 0; i < 256; i++) {
        if (freq[i] > 0) {
            printf("'%c' = %d\n", i, freq[i]);
        }
    }

    return 0;
}
```

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

Input: A string entered by the user

Process: Traverse each character and count its occurrences using an array

Output: Frequency of each character



```
Enter a string: hello
Character frequencies:
'e' = 1
'h' = 1
'l' = 2
'o' = 1

--------------------------------
Process exited after 6.352 seconds with return value
```