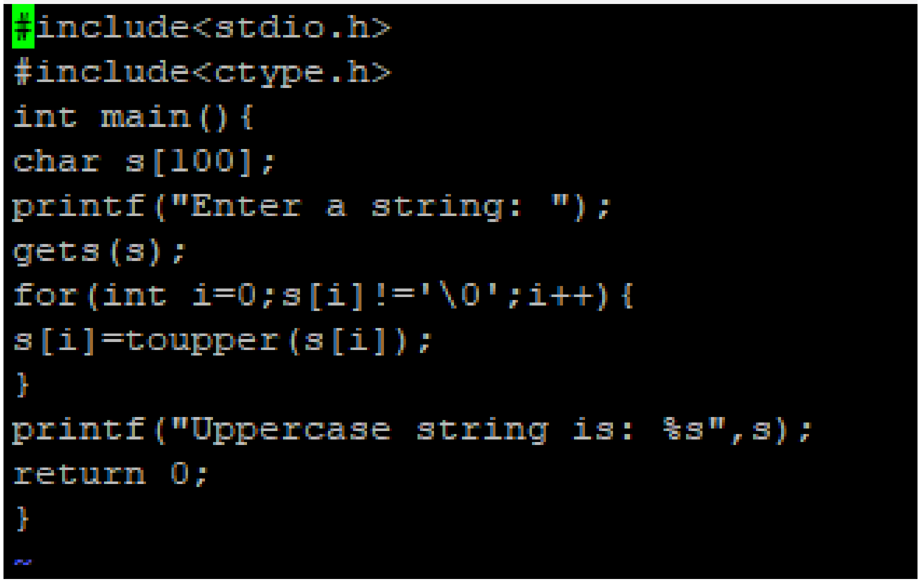
EXP NO: 1 **DATE:14/2/25**

DEVELOP A SIMPLE C PROGRAM TO DEMONSTRATE A BASIC STRING OPERATIONS

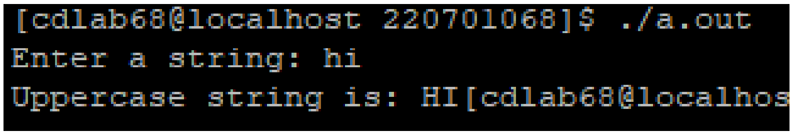
1. Input and Output

· **Question**: Modify the program to take a string input from the user and display it in uppercase.

- · **Hint**: Use the toupper function from <ctype.h> to convert characters uppercase.



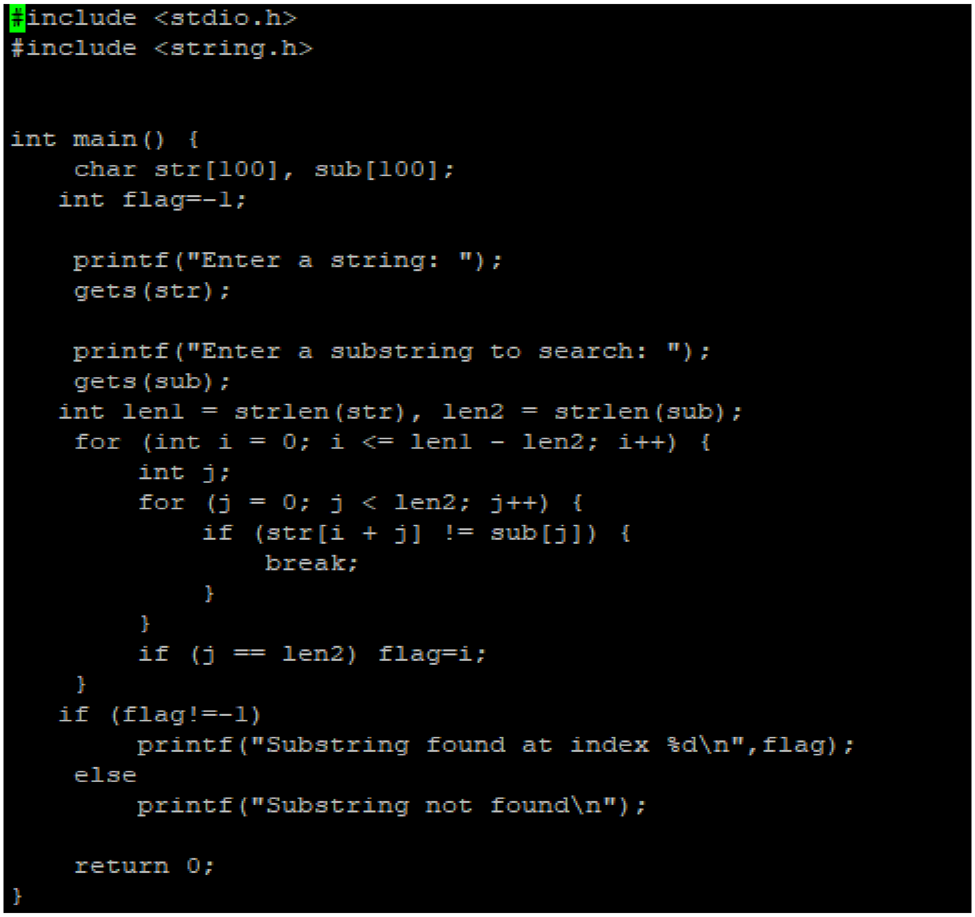
```
#include<stdio.h>
#include<ctype.h>
int main(){
char s[100];
printf("Enter a string: ");
gets(s);
for(int i=0;s[i]!='\0';i++){
s[i]=toupper(s[i]);
}
printf("Uppercase string is: %s",s);
return 0;
}
~
```



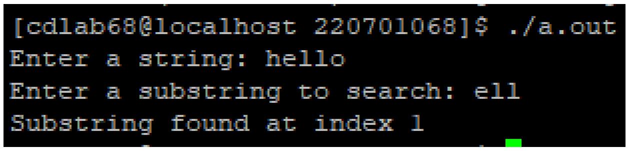
```
[cdlab68@localhost 220701068]$ ./a.out
Enter a string: hi
Uppercase string is: HI[cdlab68@localhos
```

2. String Length

Question: Write a C program to check if a given substring exists within a string without using the strstr() function. If the substring is found, print its starting index; otherwise, print "Substring not found."



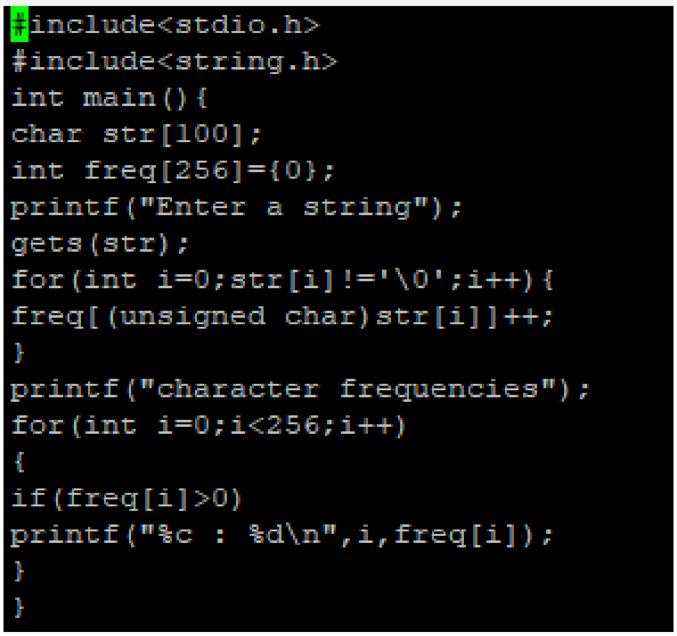
```
#include <stdio.h>
#include <string.h>


int main() {
    char str[100], sub[100];
   int flag=-1;

    printf("Enter a string: ");
    gets(str);

    printf("Enter a substring to search: ");
    gets(sub);
   int len1 = strlen(str), len2 = strlen(sub);
    for (int i = 0; i <= len1 - len2; i++) {
        int j;
        for (j = 0; j < len2; j++) {
            if (str[i + j] != sub[j]) {
                break;
            }
        }
        if (j == len2) flag=i;
    }
   if (flag!=-1)
        printf("Substring found at index %d\n",flag);
    else
        printf("Substring not found\n");

    return 0;
}
```



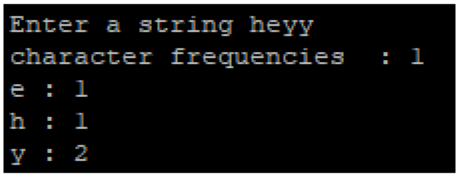
```
[cdlab68@localhost 220701068]$ ./a.out
Enter a string: hello
Enter a substring to search: ell
Substring found at index 1
```

3. String Comparison

- **Question**: Extend the program to compare two strings entered by the user and print whether they are the same.
- **Hint**: Use the strcmp function from <string.h> for comparison.



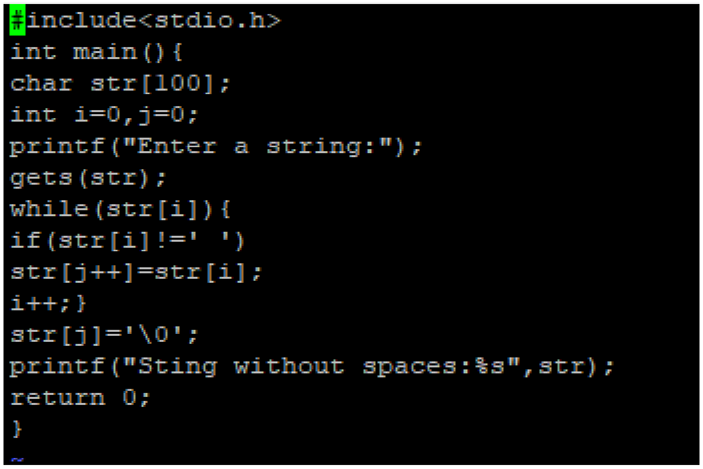
```
#include<stdio.h>
#include<string.h>
int main(){
char str[100];
int freq[256]={0};
printf("Enter a string");
gets(str);
for(int i=0;str[i]!='\0';i++){
freq[(unsigned char)str[i]]++;
}
printf("character frequencies");
for(int i=0;i<256;i++)
{
if(freq[i]>0)
printf("%c : %d\n",i,freq[i]);
}
}
```



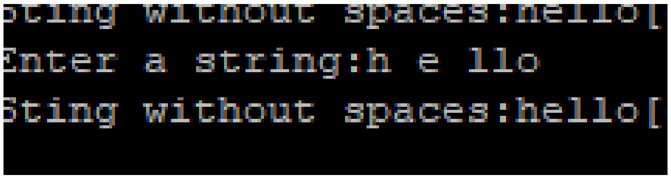
```
Enter a string heyy
character frequencies  : 1
e : 1
h : 1
y : 2
```

4. Remove Spaces

· **Question**: Write a program to remove all spaces from a string entered by the user. · **Hint**: Use a loop to copy non-space characters to a new string.



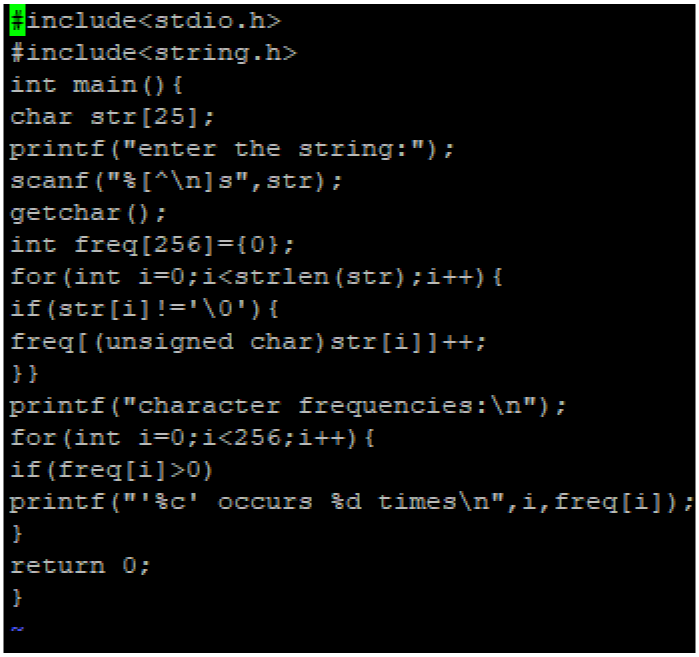
```
#include<stdio.h>
int main(){
char str[100];
int i=0,j=0;
printf("Enter a string:");
gets(str);
while(str[i]){
if(str[i]!=' ')
str[j++]=str[i];
i++;}
str[j]='\0';
printf("Sting without spaces:%s",str);
return 0;
}
~
```



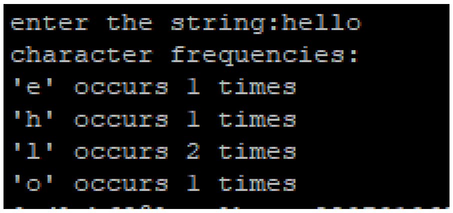
```
Sting without spaces:hello[
Enter a string:h e llo
Sting without spaces:hello[
```

5. Frequency of Characters

· **Question**: Modify the program to calculate the frequency of each character in the string. · **Hint**: Use an array of size 256 to store the count of each ASCII character.



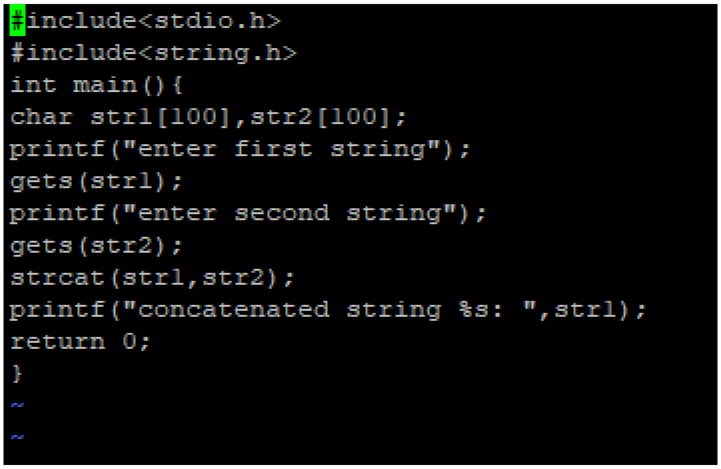
```
#include<stdio.h>
#include<string.h>
int main(){
char str[25];
printf("enter the string:");
scanf("%[^\n]s",str);
getchar();
int freq[256]={0};
for(int i=0;i<strlen(str);i++){
if(str[i]!='\0'){
freq[(unsigned char)str[i]]++;
}}
printf("character frequencies:\n");
for(int i=0;i<256;i++){
if(freq[i]>0)
printf("'%c' occurs %d times\n",i,freq[i]);
}
return 0;
}
~
```



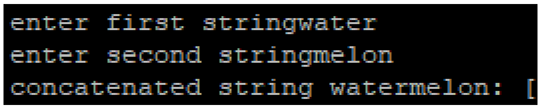
```
enter the string:hello
character frequencies:
'e' occurs 1 times
'h' occurs 1 times
'l' occurs 2 times
'o' occurs 1 times
```

6. Concatenate Strings

· **Question**: Extend the program to concatenate two strings entered by the user. · **Hint**: Use the strcat function from <string.h>.



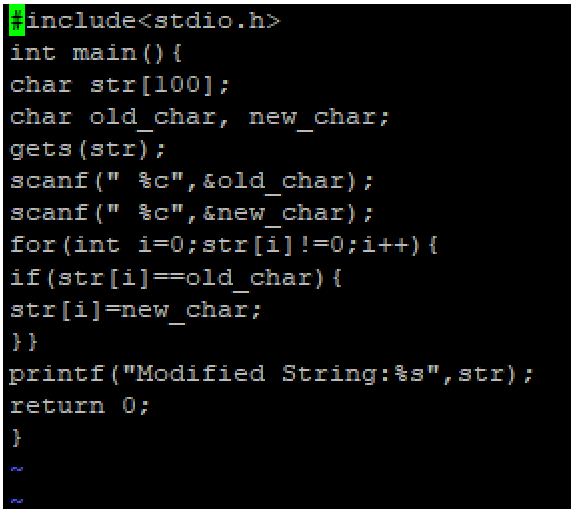
```
#include<stdio.h>
#include<string.h>
int main(){
char str1[100],str2[100];
printf("enter first string");
gets(str1);
printf("enter second string");
gets(str2);
strcat(str1,str2);
printf("concatenated string %s: ",str1);
return 0;
}
~
~
```



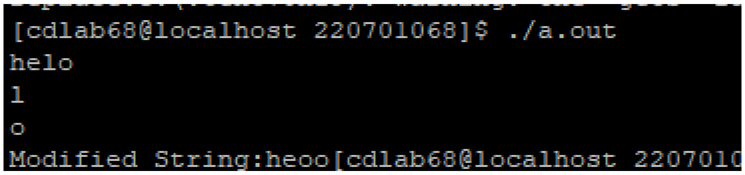
```
enter first stringwater
enter second stringmelon
concatenated string watermelon: [
```

7. Replace a Character

- **Question**: Write a program to replace all occurrences of a specific character in the string with another character.
- **Hint**: Traverse the string and replace the character conditionally in a loop.



```
#include<stdio.h>
int main(){
char str[100];
char old_char, new_char;
gets(str);
scanf(" %c",&old_char);
scanf(" %c",&new_char);
for(int i=0;str[i]!=0;i++){
if(str[i]==old_char){
str[i]=new_char;
}}
printf("Modified String:%s",str);
return 0;
}
~
~
```



```
[cdlab68@localhost 220701068]$ ./a.out
helo
l
o
Modified String:heoo[cdlab68@localhost 220701
```