EX NO-1

DEVELOP A SIMPLE C PROGRAM TO DEMONSTRATE A BASIC STRING OPERATIONS

AIM: To write a C program that takes a string input from the user and converts all its characters to uppercase using the toupper() function from the library.

PROGRAM

```c
#include <stdio.h>
#include <ctype.h>
#include <string.h>
int main() {
 char str[100];
 printf("Enter a string: ");
 fgets(str, sizeof(str), stdin);
 size_t len = strlen(str);
 if (len > 0 && str[len - 1] == '\n') {
 str[len - 1] = '\0';
 }
 for (int i = 0; str[i] != '\0'; i++) {
 str[i] = toupper((unsigned char)str[i]);
 }
 printf("Uppercase String: %s\n", str);
return 0;
}
```

OUTPUT

```
Enter a string: hello
Uppercase String: HELLO
```

KAMALI K A- 220701118

AIM: To write a C program that checks whether a given substring exists within a string without using the strstr() function. If found, print its starting index; otherwise, print "Substring not found."

PROGRAM

```
#include <stdio.h>
#include <string.h>
int findSubstring(char str[], char sub[]) {
 int strLen = strlen(str), subLen = strlen(sub);
 for (int i = 0; i <= strLen - subLen; i++) {
 int j;
 for (j = 0; j < subLen; j++) {
 if (str[i + j] != sub[j]) {
 break;
 }
 }
 if (j == subLen) {
 return i; // Found at index i
 }
 }
 return -1; // Not found
}
int main() {
 char str[100], sub[50];
 printf("Enter a string: ");
 fgets(str, sizeof(str), stdin);
 printf("Enter the substring: ");
 fgets(sub, sizeof(sub), stdin);
 str[strcspn(str, "\n")] = '\0';
 sub[strcspn(sub, "\n")] = '\0';
 int index = findSubstring(str, sub);
 if (index != -1)
 printf("Substring found at index %d\n", index);
 else
 printf("Substring not found\n");
 return 0;
}
```

OUTPUT

```
Enter a string: COMPILER DESIGN LAB
Enter the substring: LA
Substring found at index 16
```

KAMALI K A- 220701118

AIM: To write a C program that compares two strings entered by the user and determines whether they are the same.

PROGRAM

```c
#include <stdio.h>
#include <string.h>
int main() {
 char str1[100], str2[100];
 printf("Enter first string: ");
 fgets(str1, sizeof(str1), stdin);
 printf("Enter second string: ");
 fgets(str2, sizeof(str2), stdin);
 str1[strcspn(str1, "\n")] = '\0';
 str2[strcspn(str2, "\n")] = '\0';
 if (strcmp(str1, str2) == 0)
 printf("Strings are the same.\n");
 else
 printf("Strings are different.\n");
 return 0;
}
```

OUTPUT

```
Enter first string: HELLO
Enter second string: WORLD
Strings are different.
```

KAMALI K A- 220701118

AIM: To write a C program that removes all spaces from a string entered by the user.

PROGRAM

```c
#include <stdio.h>
void removeSpaces(char str[]) {
 int i, j = 0;
 for (i = 0; str[i] != '\0'; i++) {
 if (str[i] != ' ') {
 str[j++] = str[i];
 }
 }
 str[j] = '\0';
}
int main() {
 char str[100];
 printf("Enter a string: ");
 fgets(str, sizeof(str), stdin);
 removeSpaces(str);
 printf("String without spaces: %s\n", str);
 return 0;
}
```

OUTPUT

```
Enter a string: HELLO WORLD
String without spaces: HELLOWORLD
```

KAMALI K A- 220701118

AIM: To write a C program that calculates the frequency of each character in a given string.

PROGRAM

```c
#include <stdio.h>
#include <string.h>
void countFrequency(char str[]) {
 int freq[256] = {0};
 for (int i = 0; str[i] != '\0'; i++) {
 freq[(unsigned char)str[i]]++;
 }
 printf("Character Frequencies:\n");
 for (int i = 0; i < 256; i++) {
 if (freq[i] > 0) {
 printf("'%c' : %d\n", i, freq[i]);
 }
 }
}
int main() {
 char str[100];
 printf("Enter a string: ");
 fgets(str, sizeof(str), stdin);
 countFrequency(str);
 return 0;
}
```

OUTPUT

```
Enter a string: Compiler Design
Character Frequencies:
'
' : 1
' ' : 1
'C' : 1
'D' : 1
'e' : 2
'g' : 1
'i' : 2
'l' : 1
'm' : 1
'n' : 1
'o' : 1
'p' : 1
'r' : 1
's' : 1
```

KAMALI K A- 22070118

AIM: To write a C program that concatenates two strings entered by the user.

PROGRAM

```c
#include <stdio.h>
#include <string.h>
int main() {
 char str1[100], str2[50];
 printf("Enter first string: ");
 fgets(str1, sizeof(str1), stdin);
 printf("Enter second string: ");
 fgets(str2, sizeof(str2), stdin);
 str1[strcspn(str1, "\n")] = '\0';
 str2[strcspn(str2, "\n")] = '\0';
 strcat(str1, str2);
 printf("Concatenated string: %s\n", str1);
 return 0;
}
```

OUTPUT

```
Enter first string: HELLO
Enter second string: WORLD
Concatenated string: HELLOWORLD
```

KAMALI K A- 220701118

AIM: To write a C program that replaces all occurrences of a specific character in a string with another character.

PROGRAM

```c
#include <stdio.h>
void replaceChar(char str[], char oldChar, char newChar) {
 for (int i = 0; str[i] != '\0'; i++) {
 if (str[i] == oldChar) {
 str[i] = newChar;
 }
 }
}
int main() {
 char str[100], oldChar, newChar;
 printf("Enter a string: ");
 fgets(str, sizeof(str), stdin);
 printf("Enter character to replace: ");
 scanf("%c", &oldChar);
 getchar(); // Consume leftover newline character
 printf("Enter new character: ");
 scanf("%c", &newChar);
 replaceChar(str, oldChar, newChar);
 printf("Modified string: %s\n", str);
 return 0;
}
```

OUTPUT

```
Enter a string: COMPILER DESIGN
Enter character to replace: DE
Enter new character: Modified string: COMPILER
ESIGN
```

RESULT: Thus the above program takes a string input, calculates and displays its length, copies and prints the string, concatenates it with a second input string, and finally compares both strings to check if they are the same or different.

KAMALI K A- 220701118