ASSIGNMENT -11

Q1.write a program to scan string from user then scan a single

character and search it in a accepted string

```
#include <stdio.h>
#include <string.h>
int main() {
 char str[100];
 char ch;
 int found = 0;
 printf("Enter a string: ");
 scanf("%99s", str);
 printf("Enter a character to search: ");
 scanf(" %c", &ch);
 for (int i = 0; i < strlen(str); i++) {
 if (str[i] == ch) {
 found = 1;
 break;
 }
 }
 if (found) {
 printf("Character '%c' found in the string.\n", ch);
 } else {
 printf("Character '%c' not found in the string.\n", ch);
 }
 return 0;
}
```

Q2.WAP replace all occurrences of 'a' with in a string

```
#include <stdio.h>
#include <string.h>
void replace_a_with_dollar(char *str) {
 int len = strlen(str);
```

```
 for (int i = 0; i < len; i++) {
 if (str[i] == 'a') {
 str[i] = '$';
 }
 }
}
int main() {
 char str[100];
 printf("Enter a string: ");
 scanf("%99s", str);
 replace_a_with_dollar(str);
 printf("String after replacing 'a' with '$': %s\n", str);
 return 0;
}
```

Q3. WAP to remove the nth index character from a non empty string

```
#include <stdio.h>
#include <string.h>
void remove_nth_char(char *str, int n) {
 int len = strlen(str);
 if (n >= 0 && n < len) {
 for (int i = n; i < len - 1; i++) {
 str[i] = str[i + 1];
 }
 str[len - 1] = '\0';
 }
}
int main() {
 char str[100];
 int n;
 printf("Enter a string: ");
```

```
 scanf("%99s", str);
 printf("Enter the index of the character to remove: ");
 scanf("%d", &n);
 remove_nth_char(str, n);
 printf("String after removing the character: %s\n", str);
 return 0;
}
```

Q4.WAP to form a new string where the first character and the last character have been exchanged

```
#include <stdio.h>
#include <string.h>
void exchange_first_last(char *str) {
 int len = strlen(str);
 char temp = str[0];
 str[0] = str[len - 1];
 str[len - 1] = temp;
}
int main() {
 char str[100];
 printf("Enter a string: ");
 scanf("%99s", str);
 exchange_first_last(str);
 printf("String after exchanging first and last characters: %s\n", str);
 return 0;
}
```

Q5.WAP to count the number of a vowels in string

```
#include <stdio.h>
#include <string.h>
int count_vowels(char *str) {
 int count = 0;
 for (int i = 0; i < strlen(str); i++) {
```

```
 char c = tolower(str[i]);
 if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {
 count++;
 }
 }
 return count;
}
int main() {
 char str[100];
 printf("Enter a string: ");
 scanf("%99s", str);
 int vowel_count = count_vowels(str);
 printf("Number of vowels in the string: %d\n", vowel_count);
 return 0;
}
```

Q6.WAP to take a string and replace every blank space with special symbol

```
#include <stdio.h>
#include <string.h>
void replace_spaces_with_symbol(char *str, char symbol) {
 for (int i = 0; i < strlen(str); i++) {
 if (str[i] == ' ') {
 str[i] = symbol;
 }
 }
}
int main() {
 char str[100];
 char symbol;
 printf("Enter a string: ");
 scanf("%99s", str);
```

```
 printf("Enter a special symbol: ");
 scanf(" %c", &symbol);
 replace_spaces_with_symbol(str, symbol);
 printf("String after replacing spaces with symbol: %s\n", str);
 return 0;
}
```

Q7.WAP to remove the characters of odd index value in string

```
#include <stdio.h>
#include <string.h>
void remove_odd_index_chars(char *str) {
 int len = strlen(str);
 int i, j = 0;
 char temp[len + 1];
 for (i = 0; i < len; i++) {
 if (i % 2 == 0) {
 temp[j++] = str[i];
 }
 }
 temp[j] = '\0';
 strcpy(str, temp);
}
int main() {
 char str[100];
 printf("Enter a string: ");
 scanf("%99s", str);
 remove_odd_index_chars(str);
 printf("String after removing odd index characters: %s\n", str);
 return 0;
}
```

Q8.WAP to calculate the number of words present in string

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

```
#include <string.h>
int count_words(char *str) {
 int count = 0;
 int i;
 for (i = 0; i < strlen(str); i++) {
 if (str[i] == ' ') {
 count++;
 }
 }
 return count + 1; // Add 1 for the last word
}
int main() {
 char str[100];
 printf("Enter a string: ");
 scanf("%99s", str);
 int word_count = count_words(str);
 printf("Number of words in the string: %d\n", word_count);
 return 0;
}
```

Q9.WAP to take in two strings and display the larger string without using built-in functions

```
#include <stdio.h>
void larger_string(char *str1, char *str2) {
 int len1 = 0, len2 = 0;
 char *temp1 = str1;
 char *temp2 = str2;
 // Calculate length of str1
 while (*temp1 != '\0') {
 len1++;
 temp1++;
 }
```

```
 // Calculate length of str2
 while (*temp2 != '\0') {
 len2++;
 temp2++;
 }
 // Compare lengths and print the larger string
 if (len1 > len2) {
 printf("Larger string: %s\n", str1);
 } else if (len2 > len1) {
 printf("Larger string: %s\n", str2);
 } else {
 printf("Both strings are equal\n");
 }
}
int main() {
 char str1[100];
 char str2[100];
 printf("Enter first string: ");
 scanf("%99s", str1);
 printf("Enter second string: ");
 scanf("%99s", str2);
 larger_string(str1, str2);
 return 0;
}
```

Q10.Write a program to check the string is palindrom or not

```
#include <stdio.h>
#include <string.h>
int is_palindrome(char *str) {
 int start = 0;
 int end = strlen(str) - 1;
 while (start < end) {
```

```
 if (str[start] != str[end]) {
 return 0; // Not a palindrome
 }
 start++;
 end--;
 }
 return 1; // Palindrome
}
int main() {
 char str[100];
 printf("Enter a string: ");
 scanf("%99s", str);
 if (is_palindrome(str)) {
 printf("%s is a palindrome\n", str);
 } else {
 printf("%s is not a palindrome\n", str);
 }
 return 0;
}
```