Assignment

```
#include <stdio.h>
#include <string.h>
int main()
{

    char str[] = "Every dog has his day";
    char word[]="dog";
    char *pFound = NULL;
    pFound = strstr(str,word);
    printf("p=%s\n",pFound);
    printf("pfound = %p",pFound);
    return 0;
}
```

Output:

```
p=dog has his day
pfound = 0x7fffeebd4276
```

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[] = "Every dog has his day";
    char word[]="dog";
    char *pFound = NULL;
    pFound = strstr(str,word);
    if(pFound!=NULL){
        printf("%s\n",word);
    }
    else
    {
       printf("No word found\n");
    }
    printf("pfound = %p",pFound);
    return 0;
}
```

Output:

```
dog
pfound = 0x7ffc34d4eee6
```

```
#include<stdio.h>
#include <string.h>
int main()
{
    char str[] = "HI my - name is - Sanjana";
    char s[2] =" ";
    char *token = NULL;
    token = strtok(str, s);
    while(token != NULL){
        printf("token = %s \n",token);
        token = strtok(NULL, s);
    }
    return 0;
}
```

Output:

```
token = HI
token = my
token = -
token = name
token = is
token = -
token = Sanjana
```

```c
#include<stdio.h>
#include <string.h>
#include<ctype.h>
int main()
{
    char buf[100];
    int nLetters=0;
    int nDigits=0;
    int nPunct=0;
    printf("Enter an interesting string of less than %d character:\n",100);
    scanf("%s",buf);
    int i=0;
    while(buf[i])
    {
        if(isalpha(buf[i]))
        {
            ++nLetters;
        }
        else if(isdigit(buf[i]))
        {
            ++nDigits;
        }
        else if(ispunct(buf[i]))
        {
            ++nPunct;
        }
        ++i;
    }
    printf("\n Your string contained %d letters,%d digits and %d punctation
characters.\n",nLetters,nDigits,nPunct);
    return 0;
}
```

Output:

```
Enter an interesting string of less than 100 character:
HelloA1@123;..

 Your string contained 6 letters,4 digits and 4 punctation characters.
```

```
#include<stdio.h>
#include <string.h>
#include<ctype.h>
int main()
{
    char text[100];
    char substring[40];
    printf("Enter the string to searched(less than %d characters):\n",100);
    scanf("%[^\n]",text);
    printf("Enter the string(less than %d characters):\n",40);
    scanf("%s",substring);
    printf("\n First string entered:\n%s\n",text);
    printf("Second string entered:\n%s\n",substring);
    for(int i=0;(text[i]=(char)toupper(text[i]))!='\0';++i);
    for(int i=0;(substring[i]=(char)toupper(substring[i]))!='\0';++i);
    printf("The second string %s found in the first.\n",((strstr(text,substring)==NULL)?"was
not":"was"));
    return 0;
}
```

Output:

```
Enter the string to searched(less than 100 characters):
Hi Goodmorning
Enter the string(less than 40 characters):
hi

 First string entered:
Hi Goodmorning
Second string entered:
hi
The second string was found in the first.
```

```
#include<stdio.h>
#include<string.h>
void copyString_array(char to[], char from[]);
void copyString_pointer(char *to, char *from);
int main()
{
   char A[20];
   char B[50]="Sanjana";
   char choice;
   printf("Choose the copy\n");
   printf("a.Array notation\n");
   printf("p.Pointer notation\n");
   printf("Enter your choice:");
   scanf("%c",&choice);
   if(choice == 'a') {
      copyString_array(A, B);
```

```
        printf("String copied using array notation: %s\n", A);
    } else if(choice == 'p') {
        copyString_pointer(A, B);
        printf("String copied using pointer notation: %s\n", A);
    } else {
        printf("Invalid choice\n");
    }

    return 0;
}
void copyString_array(char to[],char from[])
{
    int i;
    for(i=0;from[i]!='\0';++i)
    to[i]=from[i];
    to[i]='\0';

}
void copyString_pointer(char *to,char *from)
{
    for(;*from!='\0';++from,++to){
      *to=*from;
    }
    *to = '\0';
}
```

Output 1:

```
Choose the copy
a.Array notation
p.Pointer notation
Enter your choice:a
String copied using array notation: Sanjana
```

Output 2:

```
Choose the copy
a.Array notation
p.Pointer notation
Enter your choice:p
String copied using pointer notation: Sanjana
```

Problem 1: Palindrome Checker

Problem Statement:

Write a C program to check if a given string is a palindrome. A string is considered a palindrome if it reads the same backward as forward, ignoring case and non-alphanumeric characters. Use functions like strlen(), tolower(), and isalpha().

Example:

Input: "A man, a plan, a canal, Panama"

Output: "Palindrome"

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
int main()
{
   char str[100];
   int beg=0,end=0;
   printf("Enter the string:");
   scanf("%s",str);
   while(str[end]!='\0'){
     end++;
   }
   end--;
   while(beg<end){
        if(!isalnum(str[beg]))
        {
            beg++;
            continue;
        }
        if(!isalnum(str[end]))
        {
            end--;
            continue;
        }
        if(str[beg]!=str[end] && tolower(str[beg]!=tolower(str[end])))
        {
            printf("Not a palindrome");

        }
        beg++;
        end--;

    }

    printf("palindrome");
    return 0;
}
```

Output:

```
Enter the string:"A man, a plan, a canal, Panama"
Palindrome
```

Problem 2: Word Frequency Counter

Problem Statement:

Write a program to count the frequency of each word in a given string. Use strtok() to tokenize the string and strcmp() to compare words. Ignore case differences.

Example:

Input: "This is a test. This test is simple."

Output:

```
Word: This, Frequency: 2

Word: is, Frequency: 2

Word: a, Frequency: 1

Word: test, Frequency: 2

Word: simple, Frequency: 1
```

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


int main() {
    char str[100];
    char word[20][20];
    int freq[20] = {0};
    int i, j = 0;
    int count;


    printf("Enter the string: ");
    scanf("%[^\n]", str);
    for (i = 0; str[i]; i++) {
        str[i] = tolower(str[i]);
    }
    char *token = strtok(str, " .");
    while (token != NULL)
    {
        count = 0;
        for (i = 0; i < j; i++) {
            if (strcmp(word[i], token) == 0) {
                count = 1;
                freq[i]++;
                break;
            }
        }

        if (count == 0) {
            strcpy(word[j], token);
            freq[j] = 1;
            j++;
        }
        token = strtok(NULL, " .");
    }
    for (i = 0; i < j; i++) {
        printf("Word: %s, Frequency: %d\n", word[i], freq[i]);
    }

    return 0;
}
```

Output:

```
Enter the string: This is a test. This test is simple
Word: this, Frequency: 2
Word: is, Frequency: 2
Word: a, Frequency: 1
Word: test, Frequency: 2
Word: simple, Frequency: 1
```

Problem 3: Find and Replace

Problem Statement:

Create a program that replaces all occurrences of a target substring with another substring in a given string. Use strstr() to locate the target substring and strcpy() or strncpy() for modifications.

Example:

Input:

String: "hello world, hello everyone"

Target: "hello"

Replace with: "hi"

Output: "hi world, hi everyone"

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

int main() {
    char str[100];
    char target[20], replace[20];
    char temp[100];
    char *position;
    printf("Enter the string:");
    scanf("%[^\n]",str);
    printf("Enter the target:");
    scanf("%s",target);
    printf("Enter the string to be replaced:");
    scanf("%s",replace);
    while ((position = strstr(str, target)) != NULL) {

        strncpy(temp, str, position - str);
        temp[position - str] = '\0';
        strcat(temp, replace);

        strcat(temp, position + strlen(target));

        strcpy(str, temp);
    }

    printf("Modified string: %s\n", str);
    return 0;
}
```

Output:

```
Enter the string:hello world, hello everyone
Enter the target:hello
Enter the string to be replaced:hi
Modified string: hi world, hi everyone
```

Problem 4: Reverse Words in a Sentence

Problem Statement:

Write a program to reverse the words in a given sentence. Use strtok() to extract words and strcat() to rebuild the reversed string.

Example:

Input: "The quick brown fox"

Output: "fox brown quick The"

```
#include <stdio.h>
#include <string.h>
int main() {
    char str[100];
    char *words[10];
    int i = 0;
    printf("Enter the string:");
    scanf("%[^\n]",str);
    char *token = strtok(str, " ");
    while (token != NULL) {
        words[i++] = token;
        token = strtok(NULL, " ");
    }

    printf("Reversed sentence: ");
    for (int j = i - 1; j >= 0; j--) {
        printf("%s ", words[j]);
    }
    printf("\n");

    return 0;
}
```

Output:

```
Enter the string:The quick brown fox
Reversed sentence: fox brown quick The
```

Problem 5: Longest Repeating Substring

Problem Statement:

Write a program to find the longest substring that appears more than once in a given string. Use strncpy() to extract substrings and strcmp() to compare them.

Example:

Input: "banana"

Output: "ana"

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

```
int main()
{
    char str[100];
    int maxlen=0;
    int start=0;
    printf("Enter the string:");
    scanf("%s",str);
    int n=0;
    while(str[n]!='\0')
    {
        n++;
    }
    for(int i=0;i<n;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            int k=0;
            while(i+k<n && j+k<n && str[i+k]==str[j+k])
            {
                k++;
            }
            if(k>maxlen)
            {
                maxlen=k;
                start=i;
            }
        }
    }
    if(maxlen>0)
    {
        printf("Longest repeating substring: ");
        for(int i=0;i<maxlen;i++)
        {
            printf("%c",str[start+i]);
        }
        printf("\n");
    }
    else
    return 0;
}
```

Output:

```
Enter the string:banana
Longest repeating substring: ana
```