DAY_11_ASSIGNMENT

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 string[100], str1[100], rstr[100];
    int j = 0, length;

    printf("Enter a name: ");
    scanf("%[^\n]",string);
    int length1 =strlen(string);
    for (int i = 0; i < length1; i++) {
        if (isalnum(string[i])) {
            str1[j++] = tolower(string[i]);
        }
    }

    length = strlen(str1);
    for (int i = 0; i < length; i++) {
        rstr[i] = str1[length - i - 1];
    }
    if (strcmp(str1, rstr) == 0) {
        printf("Palindrome\n");
    } else {
        printf("Not a Palindrome\n");
    }

    return 0;
}
```

2

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(){
    int count=0,i=0,k=0;
    char string[100],tem[100],word[10][10];
    scanf("%[^\n]s",string);
    strcpy(tem,string);
    char *token =strtok(tem," ");
    while(token!=NULL){
    strcpy(word[i++],token);
    token=strtok(NULL," ");
    }
    k=i;
    for(int j=0;j<k;j++){
        count=1;
        if(strcmp(word[j],"0")!=0){
        for(int l=j+1;l<k;l++){
            if(strcmp(word[j],word[l])==0){
                count++;
                strcpy(word[l],"0");
            }
        }
                printf("%s %d \n",word[j],count);
        }
    }
 }
```

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>

void main(){
    char string[30],target[10],replace[10],result[30];
    char *p;
    printf("Enter the string: ");
    scanf("%[^\n]s",string);
    printf("Target word: ");
    scanf("%s",target);
    printf("Replacing word:");
    scanf("%s",replace);
    p=strstr(string,target);
    if(p){
        int lenBeforeTarget = p - string;
        strcpy(result, string);
        result[lenBeforeTarget] = '\0';
        strcat(result, replace);
        strcat(result, p + strlen(target));

        printf("Modified string: %s\n", result);
    }
    else{
        printf("Target word is not found !");
    }
}
```

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>

void reverseWords(char *sentence) {
    char *words[100];
    int count = 0;

    char *token = strtok(sentence, " ");
    while (token != NULL) {
        words[count++] = token;
        token = strtok(NULL, " ");
    }

    char reversedSentence[1000] = "";

    for (int i = count - 1; i >= 0; i--) {
        strcat(reversedSentence, words[i]);
        if (i != 0) {
            strcat(reversedSentence, " ");
        }
    }

    printf("Reversed Sentence: %s\n", reversedSentence);
}

int main() {
    char sentence[1000];

    printf("Enter a sentence: ");
    fgets(sentence, sizeof(sentence), stdin);
    sentence[strcspn(sentence, "\n")] = '\0';

    reverseWords(sentence);

    return 0;
}
```

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>
int main(){
    char str1[100];
    char lstr[100] = "";

    printf("enter a string");
    scanf("%s",str1);
    int n= strlen(str1);
    for(int len=1;len<n;len++){
       for(int j=0;j<=n-len;j++){
            char sub1[100];
            strncpy(sub1,str1+j,len);
            sub1[len]='\0';
            for(int k=j+1;k<=n-len;k++){
                char sub2[100];
                strncpy(sub2,str1+k,len);
                sub2[len]='\0';

                if(strcmp(sub1,sub2)==0){
                    if(strlen(sub1)>strlen(lstr)){
                        strcpy(lstr,sub1);
                    }

                }
            }
       }
    }
    if(strlen(lstr)>0){
        printf("the longest substring  %s",lstr);
    }
    else{
        printf("there is no sub string");
    }
return 0;
}
```

CLASS WORKS

```
#include<stdio.h>
#include<string.h>
void copystring( char A[],char B[]);
void copystring1( char *A,char *B);

int main(){
    char A[20];
    char B[20]="anusree";
    char op;
    printf("enter the operator:");
    scanf("%c",&op);
    switch (op)
    {
    case 'a':
    copystring(A,B);
    printf("the copied first method name: %s\n",A);

        break;
    case 'b':
    copystring1(A,B);
    printf("the copied second method name: %s\n",A);
     break;

    default:
    printf("you enter the wrong operator");
        break;
    }



}
void copystring( char A[],char B[]){
    int i=0;
    for(i=0;B[i]!='\0';i++){
        A[i]=B[i];
    }
    A[i]='\0';
}
```

```
void copystring1( char *A,char *B){
    int i=0;
    for(;*B!='\0';B++,A++){
        *A=*B;
    }
    *A='\0';
}

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

    printf("strcmp(A,A)is");
    printf("%d\n",strcmp("A","A"));
    printf("strcmp(A,B)is");
    printf("%d\n",strcmp("A","B"));
    printf("strcmp(A,C)is");
    printf("%d\n",strcmp("A","C"));
    printf("strcmp(A,D)is");
    printf("%d\n",strcmp("A","D"));
    // reverse
    printf("strcmp(C,A)is");
    printf("%d\n",strcmp("C","A"));
    printf("strcmp(B,A)is");
    printf("%d\n",strcmp("B","A"));
    printf("strcmp(D,A)is");
    printf("%d\n",strcmp("D","A"));

    // apples and apple
    printf("%d\n",strcmp("apples","apple"));
    printf("%d\n",strcmp("apple","apples"));

    printf("strcmp(astounding,astro)");
    printf("%d\n",strncmp("Astounding","Astrs",5));


------------------------------------------------------------------------------------------------
#include <ctype.h>
#include<stdio.h>
#include<string.h>
int main(){
char buf[100];
int nletter =0;
int ndigit =0;
int npunct =0;

printf("enter an intersesting less than %d character",100);
scanf("%s",buf);
int i=0;
while(buf[i]!='\0'){
    if(isalpha(buf[i])){
        ++nletter;
    }
    else if(isdigit(buf[i])){
        ++ndigit;
    }
    else if(ispunct(buf[i])){
        ++npunct;
    }
    ++i;
}
printf("your string contain%d letters.%d digits and %dpunctuations
character\n",nletter,ndigit,npunct);

}
-----------------------------------------------------------------------------------------
#include<stdio.h>
#include<string.h>
int main(){
    char str1[20];
    char str2[20] ;

    strcpy(str1,"anusree");
    strncpy(str2,str1,5);
    printf("the name if str1%s  and str2%s\n ",str1,str2);
    strcat(str1,str2);
    printf("the name if str1%s  and str2%s ",str1,str2);


}
----------------------------------------------------------------------
#include<stdio.h>
#include<string.h>
int main() {
    char str[] = "anusree deepthy";
    char str1[20];
    printf("The length of str = %d\n", (int)strlen(str));
    strcpy(str1, str);
    printf("The copied value is: %s\n", str);
    printf("The str1 name is: %s\n", str1);

    return 0;
}
-----------------------------------------------------------------------------
// searching a character strch()
#include<stdio.h>
#include<string.h>
int main(){
    char str[]="the quick brown fox";
    int l= strlen(str);
    for(int i=0;i<l;i++){
        printf("str[%d] = %c ,address= %p\n" ,i,str[i],(str+i));
    }
    char ch ='q';
    char *found =NULL;
    found =strchr(str,ch);
    printf("%c\n",*found);
        printf("%p",found);

}
--------------------------------------------------------------------------
// searching a substring strstr()
#include<stdio.h>
#include<string.h>
int main(){

    char text[]="every dog has his day";
    char word[]="has";
    char *pfount =NULL;
    pfount = strstr(text,word);
    printf("The found word is: %c\n", *pfount);
    printf("%p\n",pfount);


     if (pfount != NULL) {
        printf("The found word is: %s\n", pfount);
        printf("the word is there:");
        for(int i=0;i<strlen(word);i++){
            printf("%c",pfount[i]);
        }
        printf("\n");
        for(int i=0;i<strlen(word);i++){
            printf("%p\n",(pfount+i));
        }
    } else {
        printf("Word not found.\n");
    }
}
----------------------------------------------------------------------------
#include<stdio.h>
#include<string.h>
int main(){
    char str[] ="hi my *name is* anusree";
    char s[]="*";

    char *token =NULL;
    token=strtok(str,s);
    printf("%s\n",token);
    // apply some loggic to complete it
    // to remove the - we use while loop
    while(token!=NULL){
        printf("%s\n",token);
        token=strtok(NULL,s);

    }
}
------------------------------------------------
#include <ctype.h>
#include<stdio.h>
#include<string.h>
int main(){

    char text[100];
    char substring[100];
    printf("enter the string ");
    scanf("%[^\n]",text);
    printf("enter the sub string");
    scanf("%s",substring);

    // to convert it into upper case
    for(int i=0;text[i]!='\0';i++){
        text[i]= toupper(text[i]);
        printf("%c",text[i]);
    }
    printf("\n");
    for(int i=0;substring[i]!='\0';i++){
        substring[i]= toupper(substring[i]);
        printf("%c",substring[i]);
    }
    printf("\n");
    printf("the second %s found in the first ",((strstr(text,substring)==NULL)?"was not ":"found"));
}
```

MALLOC

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int main(){
    int * str;
    int num;

    printf("enter the number of elements");
    scanf("%d",&num);
    printf("\n");

    printf("the number entered is n=%d\n",num);
    // dynamically allocated memmory from the array

    str = (int*)malloc(num*sizeof(int));

    // check wheather the memory allocated succesfully or not

    if(str == NULL){
        printf("memory not allocated");
       exit(0);
    }
    else{
        printf("memory is allocated successfully");

    }
    // populating the array
    for(int i=0;i<num;i++){
            str[i]=i+1;
        }
    // displaying the array
    for(int i=0;i<num;i++){
            printf("%d",str[i]);
        }
    free(str);
    return 0;
}
```