Embedded C Programming – Laboratory (L33+L34)

Experiment 4 – Programs on Strings and Functions

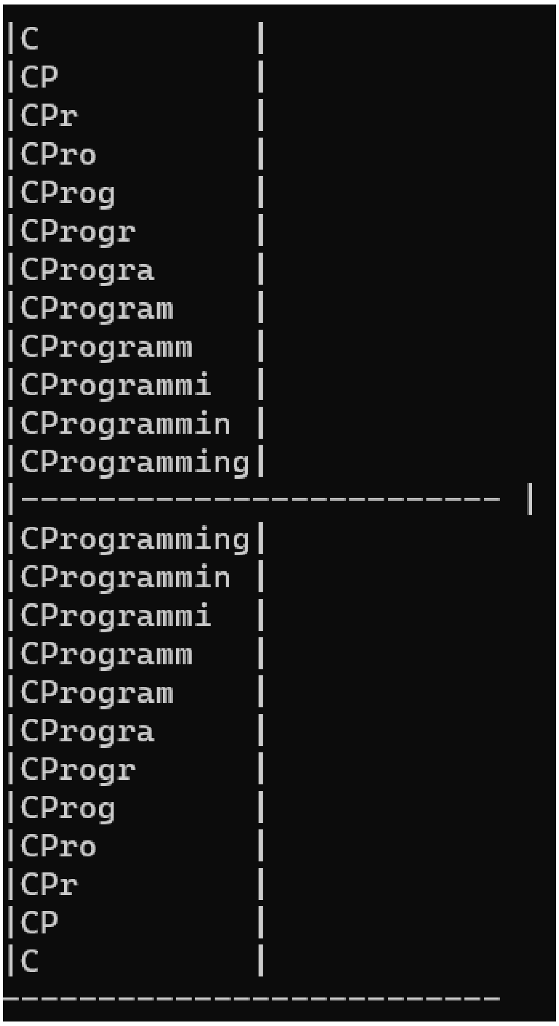
Task 1.01:

Write a program using for loop to print the output as shown.

//Pattern printing using strings

```
#include<stdio.h>
int main()
{
int c, d;
char string[] = "CProgramming";
printf("\n\n");
printf("------------------------- \n");
for( c = 0 ; c <= 11 ; c++ )
{
d = c + 1;
printf("|%-12.*s|\n", d, string);
}
printf("|------------------------- |\n");
for( c = 11 ; c >= 0 ; c-- )
{
d = c + 1;
printf("|%-12.*s|\n", d, string);
}
printf("-------------------------- \n");
}
```

Output:



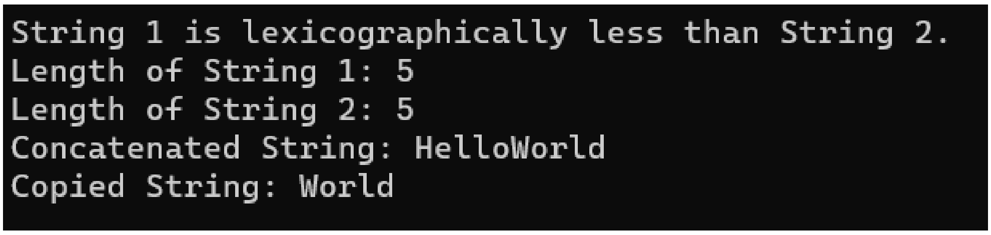
```
|C           |
|CP          |
|CPr         |
|CPro        |
|CProg       |
|CProgr      |
|CProgra     |
|CProgram    |
|CProgramm   |
|CProgrammi  |
|CProgrammin |
|CProgramming|
|-------------------------  |
|CProgramming|
|CProgrammin |
|CProgrammi  |
|CProgramm   |
|CProgram    |
|CProgra     |
|CProgr      |
|CProg       |
|CPro        |
|CPr         |
|CP          |
|C           |
--------------------------
```

Task 1.02:

//strlen, strcmp, strcat, strcpy

```
#include <stdio.h>
#include <string.h>
int main() {
    char str1[20] = "Hello";
    char str2[20] = "World";
    // Using strcmp
    int compareResult = strcmp(str1, str2);
    if (compareResult == 0) {
        printf("Strings are equal.\n");
    } else if (compareResult < 0) {
        printf("String 1 is lexicographically less than
String 2.\n");
    } else {
        printf("String 1 is lexicographically greater than
String 2.\n");
    }
    // Using strlen
    printf("Length of String 1: %lu\n", strlen(str1));
    printf("Length of String 2: %lu\n", strlen(str2));
    // Using strcat
    strcat(str1, str2);
    printf("Concatenated String: %s\n", str1);
    // Using strcpy
    strcpy(str1, "Hello");  // Reset str1
    strcpy(str1, str2);
    printf("Copied String: %s\n", str1);
    return 0;
}
```

Output

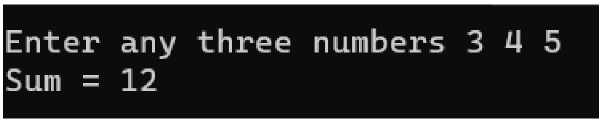


```
String 1 is lexicographically less than String 2.
Length of String 1: 5
Length of String 2: 5
Concatenated String: HelloWorld
Copied String: World
```

Task 1.03:

Calculate the sum of x, y, z using user-defined calsum function.

```c
//Sending and receiving values between functions
# include <stdio.h>
int calsum ( int x, int y, int z ) ;
int main( )
{
int a, b, c, sum ;
printf ( "Enter any three numbers " ) ;
scanf ( "%d %d %d", &a, &b, &c ) ;
sum = calsum ( a, b, c ) ;
printf ( "Sum = %d\n", sum ) ;
return 0 ;
}
int calsum ( int x, int y, int z )
{
int d ;
d = x + y + z ;
return ( d ) ;
}
```



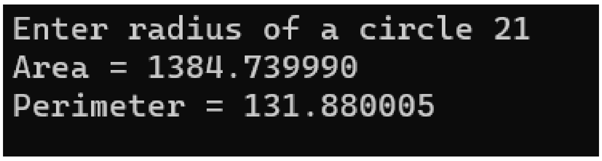
```
Enter any three numbers 3 4 5
Sum = 12
```

Task 1.04:

Write a void function to find area and perimeter of the circle and call that function inside the main function using pointers.

```c
# include <stdio.h>
void areaperi ( int, float *, float * ) ;
int main( )
{
int radius ;
float area, perimeter ;
printf ( "Enter radius of a circle " ) ;
scanf ( "%d", &radius ) ;
areaperi ( radius, &area, &perimeter ) ;//actual arguments
printf ( "Area = %f\n", area ) ;
printf ( "Perimeter = %f\n", perimeter ) ;
return 0 ;
}
void areaperi ( int r, float *a, float *p )//formal arguments
{
*a = 3.14 * r * r ;
*p = 2 * 3.14 * r ;
}
```

Output



```
Enter radius of a circle 21
Area = 1384.739990
Perimeter = 131.880005
```

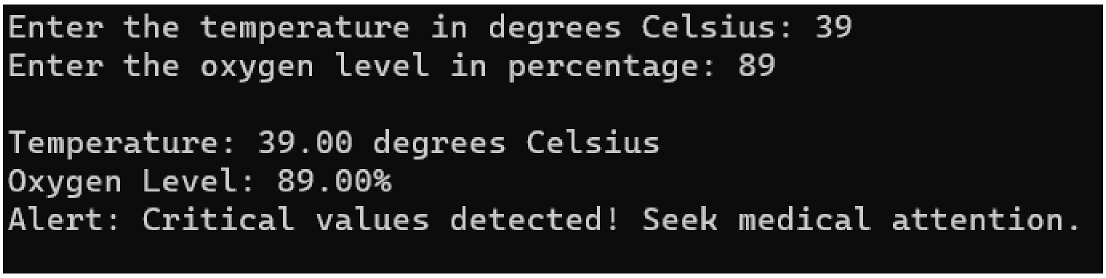
Task 1.05:

Create a function to find the temperature and oxygen level for one person. create a second user-defined function to display the critical values and provide an alert message if the values are above or below. take temperature as 37 degrees Celsius, and oxygen as 90% from pulse oximeter. (Get the input from user).

```c
#include <stdio.h>
// Function to find temperature
float findTemperature() {
    float temperature;
    printf("Enter the temperature in degrees Celsius: ");
    scanf("%f", &temperature);
    return temperature;
}

// Function to find oxygen level
float findOxygenLevel() {
    float oxygenLevel;
    printf("Enter the oxygen level in percentage: ");
    scanf("%f", &oxygenLevel);
    return oxygenLevel;
}

// Function to display critical values and provide an alert
void displayCriticalValues(float temperature, float oxygenLevel) {
    printf("\nTemperature: %.2f degrees Celsius\n", temperature);
    printf("Oxygen Level: %.2f%%\n", oxygenLevel);
    // Check for critical values and provide an alert
    if (temperature > 38.0 || oxygenLevel < 90.0) {
        printf("Alert: Critical values detected! Seek medical attention.\n");
    } else {
        printf("Values are within normal range.\n");
    }
}
int main() {
    // Get temperature and oxygen level from the user
    float temperature = findTemperature();
    float oxygenLevel = findOxygenLevel();
    // Display critical values and provide an alert
    displayCriticalValues(temperature, oxygenLevel);
    return 0;
}
```



```
Enter the temperature in degrees Celsius: 39
Enter the oxygen level in percentage: 89

Temperature: 39.00 degrees Celsius
Oxygen Level: 89.00%
Alert: Critical values detected! Seek medical attention.
```

Take home task:

Task 1.06:

s1, s2, and s3 are three string variables. Write a program to read two string constants into s1 and s2 and compare whether they are equal or not. If they are not, join them together. Then copy the contents of s1 to the variable s3. At the end, the program should print the contents of all the three variables and their lengths.