C

Assignment → 5

1. Write a Program to accept your name and print your name using string.

```
#include <stdio.h>

int main() {
   char name[100];

   printf("Enter your name: ");
   fgets(name, sizeof(name), stdin);

   printf("Your name is: %s", name);

   return 0;
}
```

2. Write a Program to reverse a string.

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

int main() {
   char str[100], rev[100];
   int i, len;

   printf("Enter a string: ");
   fgets(str, sizeof(str), stdin);

   len = strlen(str);
   for (i = 0; i < len; i++) {
      rev[i] = str[len - 1 - i];
   }
   rev[len] = '\0';
```

Python

Assignment → 5

1. Write a program to accept your name and print your name using a string.

```
name = input("Enter your name: ")
print("Your name is:", name)
```

2. Write a program to reverse a string.

```
string = input("Enter a string: ")
print("Reversed string:", string[::-1])
```

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

3. Write a Program to check whether a string is palindrome or not.

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

int main() {
    char str[100];
    int i, len, isPalindrome = 1;

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    len = strlen(str);
    for (i = 0; i < len / 2; i++) {
        if (str[i] != str[len - 1 - i]) {
            isPalindrome = 0;
            break;
        }
    }

    if (isPalindrome)
        printf("Palindrome\n");
    else
        printf("Not a palindrome\n");

    return 0;
}
```

3. Write a program to check whether a string is palindrome or not.

```
string = input("Enter a string: ")
if string == string[::-1]:
    print("Palindrome")
else:
    print("Not a palindrome")
```

4. Write a Program to find the length of string.

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

int main() {
    char str[100];
```

4. Write a program to find the length of a string.

```
string = input("Enter a string: ")
print("Length of the string:",
len(string))
```

printf("Enter a string: ");
fgets(str, sizeof(str), stdin);
printf("Length of the string: %ld\n", strlen(str));
return 0;
}	

5. Write a Program to copy one string to another string.
 	
#include <stdio.h>
#include <string.h>
 	
int main() {
char str1[100], str2[100];
 	
printf("Enter a string: ");
fgets(str1, sizeof(str1), stdin);
strcpy(str2, str1);
 	
printf("Copied string: %s\n", str2);
return 0;
}
	Write a program to copy one string to another string.
	
	string1 = input("Enter a string: ")
	string2 = string1
	print("Copied string:", string2)

6. Write a Program to concatenate two string.
 	
#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);
	6. Write a program to concatenate two strings.
	
	string1 = input("Enter first string: ")
	string2 = input("Enter second string: ")
	print("Concatenated string:", string1 + string2)

strcat(str1, str2);
 	
printf("Concatenated string: %s\n", str1);
return 0;
}	

7. Write a Program to compare two string.
 	
#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);
 	
if (strcmp(str1, str2) == 0)
printf("Strings are equal\n");
else
printf("Strings are not equal\n");
 	
return 0;
}
	7. Write a program to compare two strings.
	
	string1 = input("Enter first string: ")
	string2 = input("Enter second string: ")
	
	if string1 == string2:
	print("Strings are equal")
	else:
	print("Strings are not equal")

8. Write a Program to find the vowels in the given string. #include <stdio.h> #include <string.h> int main() { char str[100]; int i;	8. Write a program to find the vowels in the given string. string = input("Enter a string: ") vowels = "aeiouAEIOU" print("Vowels in the string:", " ".join([char for char in string if char in vowels]))

```
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    printf("Vowels in the string: ");
    for (i = 0; str[i] != '\0'; i++) {
        if (str[i] == 'a' || str[i] == 'e' ||
str[i] == 'i' || str[i] == 'o' || str[i] ==
'u' ||
            str[i] == 'A' || str[i] == 'E' ||
str[i] == 'I' || str[i] == 'O' || str[i] ==
'U') {
            printf("%c ", str[i]);
        }
    }
    printf("\n");

    return 0;
}
```

9. Write a Program to perform linear search of 5 elements taken as user input.

```
#include <stdio.h>

int main() {
    int arr[5], i, key, found = 0;

    printf("Enter 5 elements: ");
    for (i = 0; i < 5; i++)
        scanf("%d", &arr[i]);

    printf("Enter the element to
search: ");
    scanf("%d", &key);

    for (i = 0; i < 5; i++) {
        if (arr[i] == key) {
            found = 1;
            printf("Element found at index
%d\n", i);
```

9. Write a program to perform linear search on 5 elements taken as user input.

```
arr = [int(input(f"Enter element {i+1}:
")) for i in range(5)]
key = int(input("Enter the element to
search: "))

if key in arr:
    print(f"Element found at index
{arr.index(key)}")
else:
    print("Element not found")
```

```
            break;
        }
    }

    if (!found)
        printf("Element not found\n");

    return 0;
}
```

10. Write a Program to perform binary search of 5 elements taken as user input.

```
#include <stdio.h>

int main() {
    int arr[5], key, i, low, high, mid;

    printf("Enter 5 elements (in sorted order): ");
    for (i = 0; i < 5; i++)
        scanf("%d", &arr[i]);

    printf("Enter the element to search: ");
    scanf("%d", &key);

    low = 0;
    high = 4;

    while (low <= high) {
        mid = (low + high) / 2;
        if (arr[mid] == key) {
            printf("Element found at index %d\n", mid);
            return 0;
        } else if (arr[mid] < key) {
            low = mid + 1;
        } else {
```

10. Write a program to perform binary search on 5 elements taken as user input.

```
arr = [int(input(f"Enter element {i+1}: ")) for i in range(5)]
key = int(input("Enter the element to search: "))

low, high = 0, len(arr) - 1
while low <= high:
    mid = (low + high) // 2
    if arr[mid] == key:
        print(f"Element found at index {mid}")
        break
    elif arr[mid] < key:
        low = mid + 1
    else:
        high = mid - 1
else:
    print("Element not found")
```

```
            high = mid - 1;
        }
    }

    printf("Element not found\n");
    return 0;
}
```

Assignment → 6

1 Write a Program to perform bubble sort in C.

```
#include <stdio.h>

int main() {
    int arr[5], i, j, temp;

    printf("Enter 5 elements: ");
    for (i = 0; i < 5; i++)
        scanf("%d", &arr[i]);

    for (i = 0; i < 5 - 1; i++) {
        for (j = 0; j < 5 - 1 - i; j++) {
            if (arr[j] > arr[j + 1]) {
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }

    printf("Sorted array: ");
    for (i = 0; i < 5; i++)
        printf("%d ", arr[i]);

    printf("\n");
    return 0;
}
```

Assignment → 6

1. Write a program to perform bubble sort in Python.

```
arr = [int(input(f"Enter element {i+1}: ")) for i in range(5)]

for i in range(len(arr) - 1):
    for j in range(len(arr) - 1 - i):
        if arr[j] > arr[j + 1]:
            arr[j], arr[j + 1] = arr[j + 1], arr[j]

print("Sorted array:", arr)
```

2 Write a Program to perform insertion sort in C.

```c
#include <stdio.h>

int main() {
   int arr[5], i, j, key;

   printf("Enter 5 elements: ");
   for (i = 0; i < 5; i++)
      scanf("%d", &arr[i]);

   for (i = 1; i < 5; i++) {
      key = arr[i];
      j = i - 1;
      while (j >= 0 && arr[j] > key) {
         arr[j + 1] = arr[j];
         j--;
      }
      arr[j + 1] = key;
   }

   printf("Sorted array: ");
   for (i = 0; i < 5; i++)
      printf("%d ", arr[i]);

   printf("\n");
   return 0;
}
```

2. Write a program to perform insertion sort in Python.

```python
arr = [int(input(f"Enter element {i+1}: ")) for i in range(5)]

for i in range(1, len(arr)):
   key = arr[i]
   j = i - 1
   while j >= 0 and arr[j] > key:
      arr[j + 1] = arr[j]
      j -= 1
   arr[j + 1] = key

print("Sorted array:", arr)
```

4 Write a Program to perform selection sort in C.

```c
#include <stdio.h>

int main() {
   int arr[5], i, j, minIndex, temp;

   printf("Enter 5 elements: ");
   for (i = 0; i < 5; i++)
```

4. Write a program to perform selection sort in Python.

```python
arr = [int(input(f"Enter element {i+1}: ")) for i in range(5)]

for i in range(len(arr) - 1):
   min_index = i
   for j in range(i + 1, len(arr)):
      if arr[j] < arr[min_index]:
```

```
    scanf("%d", &arr[i]);

  for (i = 0; i < 4; i++) {
    minIndex = i;
    for (j = i + 1; j < 5; j++) {
      if (arr[j] < arr[minIndex])
        minIndex = j;
    }
    temp = arr[i];
    arr[i] = arr[minIndex];
    arr[minIndex] = temp;
  }

  printf("Sorted array: ");
  for (i = 0; i < 5; i++)
    printf("%d ", arr[i]);

  printf("\n");
  return 0;
}
```

Assignment → 7

1 Write a Program to add three numbers using function.

```
#include <stdio.h>

int addThreeNumbers(int a, int b, int c) {
  return a + b + c;
}

int main() {
  int num1, num2, num3, sum;

  printf("Enter three numbers: ");
  scanf("%d %d %d", &num1, &num2, &num3);
```

```
        min_index = j
    arr[i], arr[min_index] = arr[min_index], arr[i]

print("Sorted array:", arr)
```

Assignment → 7

1. Write a program to add three numbers using a function.

```
def add_three_numbers(a, b, c):
  return a + b + c

num1, num2, num3 = map(int, input("Enter three numbers: ").split())
print("Sum =", add_three_numbers(num1, num2, num3))
```

sum = addThreeNumbers(num1, num2, num3);
 	
printf("Sum = %d\n", sum);
return 0;
}	

2 Write a Program to find XY using user defined function.
 	
#include <stdio.h>
 	
long long power(int x, int y) {
long long result = 1;
while (y > 0) {
result *= x;
y--;
}
return result;
}
 	
int main() {
int x, y;
 	
printf("Enter the base (X) and exponent (Y): ");
scanf("%d %d", &x, &y);
 	
printf("%d^%d = %lld\n", x, y, power(x, y));
 	
return 0;
}
	2. Write a program to find X^Y using a user-defined function.
	
	def power(x, y):
	return x ** y
	
	x, y = map(int, input("Enter the base (X) and exponent (Y): ").split())
	print(f"{x}^{y} =", power(x, y))

3 Write a Program to find factorial of a given number using user defined functions as well as recursion function. #include <stdio.h>	3. Write a program to find the factorial of a given number using user-defined and recursive functions. def factorial(n):

```
long long factorial(int n) {
    if (n == 0 || n == 1)
        return 1;
    return n * factorial(n - 1);
}

int main() {
    int num;

    printf("Enter a number: ");
    scanf("%d", &num);

    if (num < 0)
        printf("Factorial is not defined
for negative numbers.\n");
    else
        printf("Factorial of %d is %lld\n",
num, factorial(num));

    return 0;
}
```

```
    return 1 if n == 0 or n == 1 else n *
factorial(n - 1)

num = int(input("Enter a number: "))

if num < 0:
    print("Factorial is not defined for
negative numbers.")
else:
    print(f"Factorial of {num} is
{factorial(num)}")
```

4 Write a Program to find GCD (Greatest Common Divisor) and LCM (Least Common Multiple) of two numbers using recursion.

```
#include <stdio.h>

int gcd(int a, int b) {
    if (b == 0)
        return a;
    return gcd(b, a % b);
}

int lcm(int a, int b) {
    return (a * b) / gcd(a, b);
}
```

4. Write a program to find GCD and LCM of two numbers using recursion.

```
def gcd(a, b):
    return a if b == 0 else gcd(b, a % b)

def lcm(a, b):
    return (a * b) // gcd(a, b)

num1, num2 = map(int, input("Enter
two numbers: ").split())
print(f"GCD of {num1} and {num2} is
{gcd(num1, num2)}")
print(f"LCM of {num1} and {num2} is
{lcm(num1, num2)}")
```

```
int main() {
   int num1, num2;

   printf("Enter two numbers: ");
   scanf("%d %d", &num1, &num2);

   printf("GCD of %d and %d is %d\n",
num1, num2, gcd(num1, num2));
   printf("LCM of %d and %d is %d\n",
num1, num2, lcm(num1, num2));

   return 0;
}
```

5 Write a Program to display the Fibonacci series for a given range using function.

```
#include <stdio.h>

void fibonacci_upto_n(int n) {
   int a = 0, b = 1, temp;

   printf("Fibonacci Series up to %d: ",
n);

   while (a <= n) {
      printf("%d ", a);
      temp = a + b;
      a = b;
      b = temp;
   }
   printf("\n");
}

int main() {
   int n;

   printf("Enter the value of n: ");
   scanf("%d", &n);
```

5. Write a program to display the Fibonacci series up to a given number using a function.

```
def fibonacci_upto_n(n):
   a, b = 0, 1
   print("Fibonacci Series up to", n,
end=": ")
   while a <= n:
      print(a, end=" ")
      a, b = b, a + b
   print()

n = int(input("Enter the value of n: "))
fibonacci_upto_n(n)
```

```
    fibonacci_upto_n(n);

    return 0;
}
```

6 Write a Program to check whether any use given input number is Armstrong number or not using user defined function.

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

int isArmstrong(int num) {
    int originalNum = num, sum = 0, digits = 0, remainder;

    while (originalNum != 0) {
        originalNum /= 10;
        digits++;
    }

    originalNum = num;
    while (originalNum != 0) {
        remainder = originalNum % 10;
        sum += pow(remainder, digits);
        originalNum /= 10;
    }

    return (sum == num);
}

int main() {
    int num;

    printf("Enter a number: ");
    scanf("%d", &num);

    if (isArmstrong(num))
```

6. Write a program to check whether a given number is an Armstrong number using a user-defined function.

```
def is_armstrong(num):
    digits = len(str(num))
    return num == sum(int(digit) ** digits for digit in str(num))

num = int(input("Enter a number: "))
print(f"{num} is an Armstrong number." if is_armstrong(num) else f"{num} is not an Armstrong number.")
```

```
    printf("%d is an Armstrong
number.\n", num);
  else
    printf("%d is not an Armstrong
number.\n", num);

  return 0;
}
```

7 Write a Program to check whether any use given input number is Peterson number or not using user defined function.

```
#include <stdio.h>

int factorial(int n) {
  int fact = 1;
  for (int i = 1; i <= n; i++)
    fact *= i;
  return fact;
}

int isPeterson(int num) {
  int originalNum = num, sum = 0,
digit;

  while (num > 0) {
    digit = num % 10;
    sum += factorial(digit);
    num /= 10;
  }

  return (sum == originalNum);
}

int main() {
  int num;

  printf("Enter a number: ");
```

7. Write a program to check whether a given number is a Peterson number using a user-defined function.

```
def factorial(n):
  return 1 if n == 0 else n *
factorial(n - 1)

def is_peterson(num):
  return num ==
sum(factorial(int(digit)) for digit in
str(num))

num = int(input("Enter a number: "))
print(f"{num} is a Peterson number."
if is_peterson(num) else f"{num} is
not a Peterson number.")
```

```
    scanf("%d", &num);

    if (isPeterson(num))
        printf("%d is a Peterson
number.\n", num);
    else
        printf("%d is not a Peterson
number.\n", num);

    return 0;
}
```