

## Q1

ASCII is the acronym for the American Standard Code for Information Interchange. It is a code for representing the 128 English characters as numbers, with each letter assigned a number from 0 to 127. For example, the ASCII code for uppercase A is 65. Most computers use ASCII codes to represent text, which makes it possible to transfer data from one computer to another. The standard ASCII character set uses just 7 bits for each character. There are several larger character sets that use 8 bits, which gives them 128 additional characters. The extra characters are used to represent non-English characters, graphics symbols, and mathematical symbols.

## Q2

Recursion is the process of repeating items in a self-similar way. In programming languages, when a program calls a function inside the same function, it is called a recursive call of the function. The following example calculates the factorial of a given number using a recursive function:

```
#include <stdio.h>
#include <conio.h>
int facto(int i){
    if(i <= 1){
        return 1;
    }
    return i * facto(i - 1);
}
void main() {
    int i;
    printf("Enter a number: ");
    scanf("%d", &i);
    printf("Factorial of %d is %d\n", i, facto(i));
    getch();
}
```

## Q3

A string is said to be palindrome if reverse of the string is same as string. For example, "abba" is palindrome, but "abbc" is not palindrome.

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

void isPalindrome(char str[20]) {
    int x = 0, h = strlen(str) - 1;
    while (h > x) {
        if (str[x++] != str[h--]) {
            printf("%s is Not Palindrome\n", str);
            return;
        }
    }
}
```

```

        printf("%s is palindrome\n", str);
    }

void main()
{
    char str[20];
    printf("Enter a string: ");
    gets(str);
    isPalindrome(str);
    isPalindrome("abba");
    getch();
}

```

## Q4

a) #include <stdio.h>  
#include <conio.h>

```

char xstrlenx (char str[50]) {
    int i, length = 0;
    for (i = 0; str[i] != '\0'; i++) {
        length++;
    }
    printf("The length of string is %d\n", length);
}

```

```

void main(){
    char str [50];
    printf("Enter a string: ");
    gets(str);
    xstrlenx(str);
    getch();
}

```

b) #include <stdio.h>  
#include <conio.h>

```

void xstrcpyx (char str1[50], char str2[50]){
    int c = 0;
    while (str1[c] != '\0') {
        str2[c] = str1[c];
        c++;
    }
    str2[c] = '\0';
}

```

```

void main() {
    char str1[50], str2[50];
    printf("Enter a string to copy: ");
    gets(str1);
    xstrcpyx(str1, str2);
    printf("The first string is: %s\n", str1);
    printf("After copying, the second string is: %s\n", str2);
    getch();
}

```

## Q5

Type casting refers to changing a variable of one data type into another. The compiler will automatically change one type of data into another if it makes sense. For instance, if you assign an integer value to a floating-point variable, the compiler will convert the int to a float. Casting allows you to make this type conversion explicit, or to force it when it would not normally happen. There are two types of type casting in C. They are as follows:

- **Implicit** – When the type conversion is performed automatically by the compiler without the programmer's intervention, it is known as implicit type conversion or type promotion.
- **Explicit** – The type conversion performed by the programmer by posing the data type of the expression of specific type is known as explicit type conversion. The explicit type conversion is also known as type casting.