

1. Write a c program to read characters from an array and display its ascii value, using Functions.

Template:

```
#include <stdio.h>

/* Function to display ASCII values of characters in the array */
void displayAsciiValues(char *arr, int size) {

}

int main() {
    int n;

    // Reading the number of characters
    scanf("%d", &n);

    char arr[n];

    // Reading characters into the array
    for (int i = 0; i < n; i++) {
        scanf(" %c", &arr[i]); // Note the space before %c to avoid newline issues
    }

    // Calling the function to display ASCII values
    displayAsciiValues(arr, n);

    return 0;
}
```

3 A B C	Character:A, ASCII value:65 Character:B, ASCII value:66 Character:C, ASCII value:67
4 a b c d	Character:a, ASCII value:97 Character:b, ASCII value:98 Character:c, ASCII value:99 Character:d, ASCII value:100
5 1 2 3 4 5	Character:1, ASCII value:49 Character:2, ASCII value:50 Character:3, ASCII value:51 Character:4, ASCII value:52 Character:5, ASCII value:53

2 @ #	Character:@, ASCII value:64 Character:#, ASCII value:35
6 A a 0 9 * \$	Character:A, ASCII value:65 Character:a, ASCII value:97 Character:0, ASCII value:48 Character:9, ASCII value:57 Character:*, ASCII value:42 Character:\$, ASCII value:36
3 Z z 8	Character:Z, ASCII value:90 Character:z, ASCII value:122 Character:8, ASCII value:56
7 G m 3 & ? L !	Character:G, ASCII value:71 Character:m, ASCII value:109 Character:3, ASCII value:51 Character:&, ASCII value:38 Character:?, ASCII value:63 Character:L, ASCII value:76 Character:!, ASCII value:33
1 X	Character:X, ASCII value:88
4 p Q 7 #	Character:p, ASCII value:112 Character:Q, ASCII value:81 Character:7, ASCII value:55 Character:#, ASCII value:35
5 R s T u V	Character:R, ASCII value:82 Character:s, ASCII value:115 Character:T, ASCII value:84 Character:u, ASCII value:117 Character:V, ASCII value:86

2. Write a C program to display prime numbers between intervals using function.

Template:

```
#include <stdio.h>

/* Function to check if a number is prime */
int isPrime(int num) {

    return 0; // Not a prime number

    return 1; // Prime number
}

/* Function to display prime numbers in a given range */
void displayPrimeNumbers(int start, int end) {
    printf("Prime numbers between %d and %d are:\n", start, end);
    for (int i = start; i <= end; i++) {
        if (isPrime(i)) {
            printf("%d ", i);
        }
    }
    printf("\n");
}

int main() {
    int start, end;

    // Reading the range from the user
    scanf("%d", &start);
    scanf("%d", &end);

    // Calling the function to display prime numbers in the range
    displayPrimeNumbers(start, end);

    return 0;
}
```

10 20	Prime numbers between 10 and 20 are: 11 13 17 19
1 10	Prime numbers between 1 and 10 are: 2 3 5 7

90 95	Prime numbers between 90 and 95 are:
0 2	Prime numbers between 0 and 2 are: 2
1 1	Prime numbers between 1 and 1 are:

3. Write a program in C to find the sum of the series $1!/1+2!/2+3!/3+4!/4+5!/5$ using the function by passing appropriate parameters and return values.

Template:

```
#include <stdio.h>

// Function to calculate factorial of a number and return it to sumOfSeries function
int factorial(int num) {

}

// Function to calculate the sum of the series and return it to main function
double sumOfSeries(int n) {
    double sum = 0.0;

}

int main() {
    int n;

    // Reading the value of n from the user

    scanf("%d", &n);

    // Calling the function to get the sum of the series
    double result = sumOfSeries(n);

    // Displaying the result
    printf("The sum of the series is:%.2lf\n", result);

    return 0;
}
```

1	The sum of the series is:1.00
2	The sum of the series is:2.00
5	The sum of the series is:34.00
7	The sum of the series is:874.00
10	The sum of the series is:409114.00

4. Write a C program to convert decimal numbers to binary using functions.

Template:

```
#include <stdio.h>

// Function to convert a decimal number to binary
void decimalToBinary(int decimal) {
    int binary[32]; // Array to store binary digits (up to 32 bits)
    int index = 0;

    // Edge case: If the decimal number is 0, print 0
    if (decimal == 0) {
        printf("Binary: 0\n");
        return;
    }

    // Converting decimal to binary

    // Printing the binary number in reverse order
    printf("Binary: ");
    for (int i = index - 1; i >= 0; i--) {
        printf("%d", binary[i]);
    }
    printf("\n");
}

int main() {
    int decimal;

    // Reading a decimal number from the user
```

```

scanf("%d", &decimal);

// Calling the function to convert decimal to binary
decimalToBinary(decimal);

return 0;
}

```

0	Binary: 0
1	Binary: 1
2	Binary: 10
5	Binary: 101
10	Binary: 1010
15	Binary: 1111
32	Binary: 100000
100	Binary: 1100100
255	Binary: 11111111
1023	Binary: 1111111111

5. Write a C program to pass an array of integer elements to a function and return the sum of all elements

Template:

```

#include <stdio.h>

// Function to calculate the sum of elements in an array
int sumOfArray(int arr[], int size) {
    int sum = 0;

    return sum;
}

int main() {
    int n;

```

```

// Reading the number of elements in the array
scanf("%d", &n);

int arr[n]; // Array to hold the elements

// Reading the elements of the array

for (int i = 0; i < n; i++) {
    scanf("%d", &arr[i]);
}

// Calling the function to calculate the sum
int totalSum = sumOfArray(arr, n);

// Displaying the result
printf("The sum of all elements in the array is: %d\n", totalSum);
return 0;
}

```

1 0	The sum of all elements in the array is: 0
3 1 2 3	The sum of all elements in the array is: 6
5 -1 -2 -3 -4 -5	The sum of all elements in the array is: -15
4 10 20 30 40	The sum of all elements in the array is: 100
6 -10 5 -20 15 -30 25	The sum of all elements in the array is: -15
2 2147483647 -2147483648	The sum of all elements in the array is: -1
0	The sum of all elements in the array is: 0
8 0 0 0 0 0 0 0 0	The sum of all elements in the array is: 0

10 1 -1 2 -2 3 -3 4 -4 5 -5	The sum of all elements in the array is: 0
7 100 200 300 400 500 600 700	The sum of all elements in the array is: 2800