Que.1 Write a C program to check whether a number is prime or not.

```
#include <stdio.h>
int main()
{
   int i, num, p = 0;
   printf("Enter number: ");
   scanf("%d", &num);
   for (i = 2; i <= num / 2; i++)
   {
      if (num % i == 0)
      {
         p++;
         break;
      }
   }
   if (p == 0 && num != 1)
   {
      printf("%d Prime number", num);
   }
   else
   {
      printf("%d not a Prime number", num);
   }
   return 0;
}
```

Que.2 Implement a C program that reads a string from the user and counts the number of vowels and consonants in it.

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

int main() {
   char str[100];
   int i, len, vowel=0, cons=0;

   printf("\n\nCount total number of vowel or consonant :\n");

   printf("Enter the string : ");
   scanf("%s",str);

   len = strlen(str);
   for (i = 0; i < len; 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') {
         vowel++;
      }

      else if ((str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z')) {
         cons++;
      }
   }

   printf("\nThe total vowel : %d\n", vowel);
   printf("The total consonant: %d\n\n", cons);

         return 0;
}
```

Que.3 Write a C program to calculate the sum of the digits of an integer number entered by the user.

```
#include<stdio.h>
 int main()
{
int n,sum=0,m;
printf("Enter a number:");
scanf("%d",&n);
while(n>0)
{
m=n%10;
sum=sum+m;
n=n/10;
}
printf("Sum is=%d",sum);
```

```
return 0;
}
```

Que.4 Develop a C program that finds the transpose of a given matrix.

```
#include <stdio.h>
int main() {
  int a[10][10] = { {1, 2, 3}, {4, 5, 6},{7,8,9} };
  int transpose[10][10], r=3, c=3;

  printf("\nYour matrix: \n");
  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    printf("%d  ", a[i][j]);
    if (j == c - 1)
    printf("\n");
  }

  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    transpose[j][i] = a[i][j];
  }

  printf("\nTranspose of the matrix:\n");
  for (int i = 0; i < c; ++i)
  for (int j = 0; j < r; ++j) {
    printf("%d  ", transpose[i][j]);
    if (j == r - 1)
    printf("\n");
  }
  return 0;
}
```

Que.5 Write a C program to merge two sorted arrays into a single sorted array.

```
// Online C compiler to run C program online
#include <stdio.h>
#define MAX_SIZE 100
int main()
{
    int arr1[MAX_SIZE], arr2[MAX_SIZE], merged[MAX_SIZE * 2];
    int size1, size2, i = 0, j = 0, k = 0;
    printf("Enter size of first array: ");
    scanf("%d", &size1);
    printf("Enter elements of first array:\n");
    for (i = 0; i < size1; i++)
        scanf("%d", &arr1[i]);
    printf("Enter size of second array: ");
    scanf("%d", &size2);
    printf("Enter elements of second array:\n");
    for (i = 0; i < size2; i++)
        scanf("%d", &arr2[i]);
    i = 0;
    j = 0;
```

```
    k = 0;
    while (i < size1 && j < size2)
    {
        if (arr1[i] < arr2[j])
            merged[k++] = arr1[i++];
        else
            merged[k++] = arr2[j++];
    }
    while (i < size1)
        merged[k++] = arr1[i++];
    while (j < size2)
        merged[k++] = arr2[j++];

    printf("Merged array:\n");
    for (i = 0; i < size1 + size2; i++)
        printf("%d ", merged[i]);
    printf("\n");

    return 0;
}
```

Que.6 Create a C program that calculates the sum and average of elements in an array.

```
#include<stdio.h>

int main()
{
 float  sum=0, avg;
 int i, n=5;
 float a[100]={5,4,3,2,1};

 for(i=0; i< n; i++)
 {
  sum = sum + a[i];
 }

 avg = sum/n;

 printf("Sum is %f\n", sum);
 printf("Average is %f", avg);

 return 0;
}
```

Que.7 Develop a C program that checks whether a given string is a palindrome without using built-in functions.

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

int main()
{
```

```
   char str[10];
   int i, len, flag = 0;

printf ("Enter the String\n");
scanf("\n%s",&str);

 len = strlen(str);

   for (i = 0; i < len; i++)
   {
         // Checking if string is palindrome or not
      if (str[i] != str[len - i - 1]) {
         flag = 1;
         break;
      }
   }

   if (flag)
      printf("%s is not palindrome", str);
   else
      printf("%s is palindrome", str);

   return 0;
}
```

Que.8 Write a C program to find the second largest element in an array of integers.

```
#include <stdio.h>

void main() {
   int arr1[50]={2,9,7,6,5};
   int n=5, i=0, j = 0, lrg=0, lrg2nd=0;

   for (i = 0; i < n; i++) {
      if (lrg < arr1[i]) {
         lrg = arr1[i];
         j = i;
      }
   }

   for (i = 0; i < n; i++) {
      if (i == j) {
         i++;
         i--;
      } else {
         if (lrg2nd < arr1[i]) {
```

```
            lrg2nd = arr1[i];
        }
      }
    }

   printf("The Second largest element in the array is :  %d \n\n", lrg2nd);
}
```

Que.9 Implement a C program that multiplies two matrices.

```
#include <stdio.h>

int main() {

   int matrix1[3][4] = {
      {1,2,3,0},
      {4,5,6,0},
      {7,8,9,0}
   };
   int matrix2[3][4] = {
      {11,22,33,0},
      {44,55,66,0},
      {77,88,99,0}
   };
   int len_of_row = sizeof(matrix1)/sizeof(matrix1[0]);
   int len_of_col = (sizeof(matrix1[0])/sizeof(matrix1[0][0]));

   printf("Matrix-1:\n");
   for(int row = 0; row<len_of_row; row++){
      for(int col = 0; col<len_of_col; col++){
         printf("%d ", matrix1[row][col]);
      }
      printf("\n");
   }

   printf("Matrix-2:\n");
   for(int row = 0; row<len_of_row; row++){
      for(int col = 0; col<len_of_col; col++){
         printf("%d ", matrix2[row][col]);
      }
      printf("\n");
   }
   printf("Matrix-1 + Matrix-2:\n");
   for(int row = 0; row<len_of_row; row++){
      for(int col = 0; col<len_of_col; col++){
         printf("%d ",matrix1[row][col] * matrix2[row][col]);
      }
      printf("\n");
   }
```

```
    return 0;
}
```

Que.10 Write a C program to count the frequency of each character in a given string.

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

int main()
{
   char S[100];
   int i = 0;
   printf("Enter String\n");
   scanf("%s",&S);
   int freq[26] = { 0 };

   while (S[i] != '\0') {
   freq[S[i] - 'a']++;

     i++;
   }
    for (int i = 0; i < 26; i++)
   {
     if (freq[i] != 0)
      {
         printf("%c - %d\n", i + 'a', freq[i]);
      }
   }
 }
```

Que.11 Create a C program that reads an array of integers from the user and removes duplicate elements from the array.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
   int n, count = 0;
   printf("Enter number of elements in the array: ");
   scanf("%d", &n);
   int arr[n], temp[n];
   if(n==0)
   {
      printf("No element inside the array.");
      exit(0);
   }
   printf("Enter elements in the array: ");
   for (int i = 0; i < n; i++)
   {
      scanf("%d", &arr[i]);
   }
```

```
    printf("\nArray Before Removing Duplicates: ");
    for (int i = 0; i < n; i++)
       printf("%d ", arr[i]);

    // To store unique elements in temp after removing the duplicate elements
    for (int i = 0; i < n; i++)
    {
       int j;
       for (j = 0; j < count; j++)
       {
         if (arr[i] == temp[j])
           break;
       }
       if (j == count)
       {
         temp[count] = arr[i];
         count++;
       }
    }

    printf("\nArray After  Removing Duplicates: ");
    for (int i = 0; i < count; i++)
        printf("%d ", temp[i]);

    return 0;
}
```

Que.12 Write a C program to find the factorial of a number using both iterative and recursive approaches.

```
#include <stdio.h>

int factorialUsingRecursion(int n)
{
   if (n == 0)
      return 1;
   return n * factorialUsingRecursion(n - 1);
}

int factorialUsingIteration(int n)
{
   int res = 1, i;
    for (i = 2; i <= n; i++)
      res *= i;
    return res;
}


int main()
{
   int num;

   printf("Enter the Factorial Number\n") ;
   scanf("%d=", &num);

   printf("Factorial of %d using Recursion is: %d\n", num,
```

```
        factorialUsingRecursion(num));

    printf("Factorial of %d using Iteration is: %d", num,
        factorialUsingIteration(num));

    return 0;
}
```

Que.13 Create a C program that implements a basic calculator using functions for addition, subtraction, multiplication, and division.

```
#include <stdio.h>

int main() {
    int num1,num2,op,total;
    printf("Enter the first operand\n");
    scanf("%d",&num1);

    printf("Enter the secon operand\n");
    scanf("%d",&num2);

    printf("select the operation\nPress 1 => Addition\nPress 2 => Subtraction\nPress 3
=> Multiplication\nPress 4 => Division\nPress 5 => Modulo");
    scanf("%d",&op);

    if(op==1){
        total=num1+num2;
        printf("Your Answer is = %d", total);
    }
    if(op==2){
        total=num1-num2;
        printf("Your Answer is = %d", total);
    }
    if(op==3){
        total=num1*num2;
        printf("Your Answer is = %d", total);
    }
    if(op==4){
        total=num1/num2;
        printf("Your Answer is = %d", total);
    }
    if(op==5){
        total=num1%num2;
        printf("Your Answer is = %d", total);
    }


    return 0;
}
```

Que.14 Write a C program to reverse the words in a given sentence without using any library functions.

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main(){
  char string[20],temp;
```

```
  int i,length;
  printf("Enter String : ");
  scanf("%s",string);
  length=strlen(string)-1;
  for(i=0;i<strlen(string)/2;i++){
    temp=string[i];
    string[i]=string[length];
    string[length--]=temp;
  }
  printf(" Reverse string :%s",string);
  getch();
}
```

Que.15 Create a C program that finds the largest and smallest elements in a matrix.

```
#include <stdio.h>

int main() {
    int arr[] = {10, 5, 8, 20, 3, 15};
    int n = sizeof(arr) / sizeof(arr[0]);
    int smallest = arr[0];
    int largest = arr[0];

    for (int i = 1; i < n; i++) {
        if (arr[i] < smallest) {
            smallest = arr[i];
        }
        if (arr[i] > largest) {
            largest = arr[i];
        }
    }

    printf("Smallest number: %d\n", smallest);
    printf("Largest number: %d\n", largest);

    return 0;
}
```

Que.16 Write a C program to convert a given string to lowercase without using built-in functions.

```
#include <stdio.h>

void lower_case(char arr[], int len){

        for(int pos = 0; pos<len;pos++){

        if ((arr[pos] >= 'A') && (arr[pos] <='Z')){
            printf("%c", arr[pos] + 32);
            continue;
        }
        printf("%c", arr[pos]);
    }
}

int main() {
    char name[] = "PyThoN cOde";

    int len = sizeof(name)/sizeof(name[0]);
    lower_case(name, len);
```

```
    return 0;
}
```

Que.17 Develop a C program that takes a string as input and removes all white spaces.

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

int main()
{
char str[50];
int i, j;

printf("Enter a string: ");
gets(str);

i = 0;
j = 0;
while (str[i] != '\0')
{
if (str[i] != ' ')
{
str[j] = str[i];
j++;
}
i++;
}
str[j] = '\0';

printf("String after removing all the white spaces: %s", str);

return 0;
}
```

Que.18 Create a C program that checks if a given year is a leap year.

```
#include <stdio.h>

int main() {
    int year;
    printf("Enter a year: ");
    scanf("%d", &year);
    if (year % 4 == 0) {
        if (year % 100 != 0 || year % 400 == 0) {
            printf("%d is a leap year.\n", year);
        } else {
            printf("%d is not a leap year.\n", year);
        }
    } else {
        printf("%d is not a leap year.\n", year);
    }

    return 0;
}
```

Que.19 Write a C program to find the length of a string without using any built-in functions.

```
#include <stdio.h>
int main()
```

```
{
    char str[100];
    int i,len=0;

    printf("Enter a string: \n");
    scanf("%s",str);
    for(i=0; str[i]!='\0'; i++)
    {
        len++;
    }

    printf("\nLength of input string: %d",len);
     return 0;
}
```

Que.20 Implement a C program that converts temperature from Celsius to Fahrenheit using the formula F = (C × 9/5) + 32.

```
#include <stdio.h>

   int main()
{
   float celsius, fahrenheit;
   printf("Enter temperature in Celsius: ");
   scanf("%f", &celsius);
   fahrenheit = (celsius * 9 / 5) + 32;
   printf("Temperature in Fahrenheit: %.2f\n", fahrenheit);
   return 0;
}
```