Panipat Institute of Engineering & Technology, Samalkha, Panipat

Applied Sciences and Humanities (ASH) Department

Course Title: Problem Solving Using 'C' Lab
Course Code: BT-CSE-171A
Academic Year: 2025-2026

Submitted to:
Mr. Gagan Dua
Assistant Professor
(B2)
Dept. CSE

Submitted By:
Amrita Malik
Roll No.28250301

Index

S. No.	Title of the Practical	Date	Teacher Signature
1	Write a program to find area of a circle.		
2	Write a program to swap two numbers with and without using a third variable.		
3	Write a program to find the sum of individual digits of a positive integer.		
4	Write a program to generate all the prime numbers between 1 and *n*, where n is the input given by the user.		
5	Write a function to generate Pascal's triangle.		
6	Write a program to find the roots of a quadratic equation.		
7	Program to calculate the sum of first n natural numbers.		
8	Write a program to print different pyramid patterns.		
9	Write programs to find the factorial of a given integer by using both recursive and non-recursive functions.		
10	Write a program to generate the first n terms of the Fibonacci sequence		
11	Write a program for addition of two matrices.		
12	Write a program to calculate the following series without pow() function x - x3/3! +x5/5! - x7/7! ... xn/n!		
13	Write a program for calculating transpose of a matrix.		
14	Write a program for matrix multiplication by checking compatibility.		
15	Write a program to concatenate two strings.		
16	Write a program to implement Structure for storing information about an employee.		
17	Write a program to implement Union.		
18	Write a program to print the element of array using pointers.		

19	Write a program to print the elements of a structure using pointers.		
20	Write a program to explore malloc and calloc.		
21	Write a program to create a file.		
22	Write a program which copies one file to another.		
23	Write a program that counts the number of characters and number of lines in a text file.		
24	Write a program that changes every 5th character of data file into uppercase.		

Practical No. 1

Aim: Write a program to find area of circle.

Software Requirements: visual studio code

Code:

```
#include <stdio.h>
int main()
{
   printf("======================\n");
   printf("Performed by: 28250301,B2\n");
   printf("======================\n");
   float radius, area;
   // Ask the user to enter the radius
   printf("Enter the radius of the circle: ");
   scanf("%f", &radius);
  // Calculate the area of circle
   area = PI * radius * radius;
  //display the final result.
   printf("Area of the circle = %.2f\n", area);
   return 0;
 }
```

OUTPUT:

```
====================
Performed by: 28250301,B2
====================
Enter the radius of the circle: 5
Area of the circle = 78.54
PS C:\Users\DELL\OneDrive\Desktop\amrita project\C TUTORIALS>
```

Practical No. 2 (a)

Aim: Write a program to swap two numbers using a third variable

Software Requirements: Visual Studio Code

Code:

```
# include<stdio.h>
int main()
{
  printf("====================\n");
  printf("PERFORMED BY: 28250301, B2\n");
  printf("====================\n");

  int a,b,c;
  printf("ENTER THE VALUE OF a: ");
  scanf("%d",&a);
  printf("ENTER THE VALUE OF b: ");
  scanf("%d",&b);
  printf("BEFORE SWAPPINGVALUE OF  a: %d and b: %d\n",a, b);
  c=a;
   a=b;
  b=c;
  printf("AFTER SWAPPING VALUE OF  a: %d and b: %d\n",a, b);
  return 0;
}
```

OUTPUT:

```
====================
PERFORMED BY: 28250301, B2
====================
ENTER THE VALUE OF a: 123
ENTER THE VALUE OF b: 456
BEFORE SWAPPINGVALUE OFa: 123 and b: 456
AFTER SWAPPING VALUE OFa: 456 and b: 123
PS C:\Users\DELL\OneDrive\Desktop\amrita project\C TUTORIALS>
```

Practical No. 2 (b)

Aim: Write a program to swap two numbers without using a third variable

Software Requirements: Visual Studio Code

Code:

```
#include <stdio.h>
int main()
{
 printf("====================\n");
 printf("PERFORMED BY: 28250301, B2\n");
 printf("====================\n");
 int a, b;
 printf("ENTER THE VALUE OF a: ");
 scanf("%d", &a);
 printf("ENTER THE VALUE OF b: ");
 scanf("%d", &b);
 printf("BEFORE SWAPPING VALUE OF a: %d and b: %d\n", a, b);
  a = a + b;
  b = a - b;
  a = a - b;
 printf("AFTER SWAPPING VALUE OF a: %d and b: %d\n", a, b);
 return 0;
}
```

OUTPUT:-

```
====================
PERFORMED BY: 28250301, B2
====================
ENTER THE VALUE OF a: 4
ENTER THE VALUE OF b: 6
BEFORE SWAPPING VALUE OF a: 4 and b: 6
AFTER SWAPPING VALUE OF a: 6 and b: 4
```

Practical No.3

Aim: Write a program to find sum of individual digits of positive integer.

Software Requirements: Visual Studio Code

Code:

```
#include <stdio.h>
int main()
 {
   printf("==========================\n");
   printf("PERFORMED BY: 28250301,B2\n");
   printf("==========================\n");
   int a, b = 0, digit;
   printf("ENTER THE NUMBER: ");
   scanf("%d", &a);
   while(a>0)
{
  digit = a%10; b =b+digit;
  a =a/10;
}
  printf("THE SUM IS : %d",b);
  return 0;
}
```

OUTPUT:

```
==========================
PERFORMED BY: 28250301,B2
==========================
ENTER THE NUMBER: 35
THE SUM IS : 8
PS C:\Users\DELL\OneDrive\Desktop\amrita project\C TUTORIALS>
```

Practical No.4

Aim: write a program to generate all the prime numbers between 1 and n , where n is the input given by the user .

Software Requirements: **Visual Studio Code**

Code:

```
#include<stdio.h>
int main()
{
 printf("========================\n");
 printf("PERFORMED BY: 28250301, B2\n");
 printf("========================\n");
 int n, i, j, count = 0;
 printf("ENTER THE NUMBER:");
 scanf("%d", &n);
 printf("THE PRIME  NUMBERS ARE:\n");
 for(i = 2; i < n; i = i+1)
{
 int flag = 1;
 for (j = 2; j < i; j = j + 1)
{
  if (i%j == 0)
  {
    flag = 0;
    break;
  }
 }
  if ( flag == 1)
  printf("%d\n", i);
 }
}
```

Output:

```
========================
PERFORMED BY: 28250301, B2
========================
ENTER THE NUMBER:12
THE PRIME NUMBERS ARE:
2
3
5
7
11
PS C:\Users\DELL\OneDrive\Desktop\amrita project\C TUTORIALS>
```

Practical No.5

Aim: Write a function to generate Pascal's triangle.

Software required: **Visual Studio Code**

Code:

```
#include<stdio.h>
int main()
{
  int i,j,k,n,value;
  printf("======================\n");
  printf("Performed by: 28250301,B2\n");
  printf("======================\n");

  printf("Enter the number of rows: ");
  scanf("%d",&n);
  for(i=0;i<n;i++){
  for(k=n-i;k>0;k--)
  printf(" ");
  value=1;
  for(j=0;j<=i;j++){
  printf("%d ",value);
  value=value*(i-j)/(j+1);
}
  printf("\n");
}
return 0;
}
```

OUTPUT:

```
======================
Performed by: 28250301,B2
======================
Enter the number of rows: 6
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
```

Practical No. 6

Aim: Write a program to find the roots of a quadratic equation **.**

Software Requirements: Visual Studio Code

Code :

```
#include<stdio.h>
int main()
{
  printf("======================\n");
  printf("Performed by: 28250301,B2\n");
  printf("======================\n");
  float a, b, c, d, root1, root2;
  printf("Enter the value of a: ");
  scanf("%f",&a);
  printf("Enter the value of b: ");
  scanf("%f",&b);
  printf("Enter the value of c: ");
  scanf("%f",&c);
  d = b*b-4*a*c;
  root1 = -b  / (2*a);
  root2 = -b  / (2*a);
  printf("root1 = % .2f \n",root1);
  printf("root2 = % .2f \n",root2);
}
```

Output:

```
======================
Performed by: 28250301,B2
======================
Enter the value of a: 2
Enter the value of b: 3
Enter the value of c: 4
root1 = -0.75
root2 = -0.75
```

Practical No. 7

Aim: Program to calculate the sum of first n natural numbers.

Software Requirements: Visual Studio Code

Code :

```
#include <stdio.h>
int main()
{
  printf("======================\n");
  printf("PERFORMED BY: 28250301, B2\n");
  printf("======================\n");
  int n, sum = 0;
  printf("ENTER THE NUMBER: “);
  scanf("%d", &n);
  for(int i=1; i<=n; i = i + 1)
  {
    sum = sum + i;
   }
     printf("THE SUM IS: %d", sum);
}
```

Output:

```
======================
PERFORMED BY: 28250301, B2
======================
ENTER THE NUMBER:24
THE SUM IS: 300
PS C:\Users\DELL\OneDrive\Desktop\amrita project\C TUTORIALS>
```

Practical No. (8.1)

Aim: Write a program to print different pyramid patterns.

Software Requirements: Virtual Studio Code

Code:

```
#include<stdio.h>
 int main()
{
  printf("======================\n");
  printf("PERFORMED BY: 28250301, B2\n");
  printf("======================\n");
  int i,rows;
  printf("Put the no of rows: ");
  scanf("%d",&rows);
  for(i=1;i<=rows;i++)
  {
   for (int j=1; j<=i; j++)
    {
      printf("*");
    }
      printf("\n");
   }
}
```

Output:

```
======================
PERFORMED BY: 28250301, B2
======================
Put the no of rows: 5
*
**
***
****
*****
```

Practical No. (8.2)

Aim: Write a program to print different pyramid patterns.

Software Requirements: Virtual Studio Code

Code:

```
#include<stdio.h>
int main()
{
  printf("======================\n");
  printf("PERFORMED BY: 28250301, B2\n");
  printf("======================\n");

  int i, j, rows, k;
  printf("Enter the no of rows: ");
  scanf("%d", &rows);
  for(i=1;i<=rows;i++)
  {
    for ( j=i; j<=rows; j++)
   {
    printf(" ");
   }
    for (k=1;k<=i; k++)
  {
    printf("*");
  }
    printf("\n");
  }
}
```

Output:

```
======================
PERFORMED BY: 28250301, B2
======================
Enter the no of rows: 5
      *
     **
    ***
   ****
  *****
```

Practical No. (8.3)

Aim : Write a program to print different pyramid patterns.

Software Requirements: Virtual Studio Code

Code:

```
#include<stdio.h>
int main()
{
  printf("======================\n");
  printf("PERFORMED BY: 28250301, B2\n");
  printf("======================\n");

  int i, j, k, rows;
  printf("Enter the no of rows:");
  scanf("%d", &rows);
  for (i=1;i<=rows;i++)
  {
    for( j=1; j<=rows-i; j++)
  {
     printf(" ");
   }
    for ( k = 1;k<=2*i-1;k++)
  {
    printf("*");
   }
    printf("\n");
   }
}
```

Output:

```
======================
PERFORMED BY: 28250301, B2
======================
Enter the no of rows:5
    *
   ***
  *****
 *******
*********
```

Practical No. 9(a)

Aim: Write programs to find the factorial of a given integer by using recursive functions.

Software Requirements: Visual Studio Code

Code:

```
#include<stdio.h>
int factorial(int n)
 {
  if(n == 0)
{
  return 1;
}
 else
 {
 return  n*factorial(n-1);
 }
}
 void main()
{
  int n,r;
  printf("======================\n");
  printf("Performed by: 28250301,B2\n");
  printf("======================\n");
  printf("Enter number to find it's factorial = ");
  scanf("%d",&n);
  r = factorial(n);
  printf("Factorial of %d = %d",n,r);
}
```

OUTPUT:-

```
======================
Performed by: 28250301,B2
======================
Enter number to find it's factorial = 6
Factorial of 6 = 720
```

Practical No. 9(b)

Aim: Write programs to find the factorial of a given integer by using non-recursive functions.

Software Requirements: Visual Studio Code

Code:

```
#include <stdio.h>
long long fact(int n);
int main()
{
  printf("======================\n");
  printf("Performed by: 28250301,B2\n");
  printf("======================\n");
   int num;
   printf("Enter a number : ");
   scanf("%d", &num);
   printf("The factorial of %d is %lld \n", num, fact(num));
   return 0;
}
long long fact(int n)
{
   long long fact = 1;
   for (int i = 1; i <= n; i++)
   {
      fact *= i;
   }
   return fact;
}
```

Output:

```
======================
Performed by: 28250301,B2
======================
Enter a number : 5
The factorial of 5 is 120
PS C:\Users\codew\c lanuage\arrays>
```

Practical No.10

Aim: Write a program to generate the first n terms of the Fibonacci sequence.

Software Requirements: Virtual Studio Code

Code:

```
#include<stdio.h>
int main()
 {
  printf("======================\n");
  printf("Performed by: 28250301,B2\n");
  printf("======================\n");

  int n, first =0,second=1,temp,i;
  printf("Enter number of terms... ");
  scanf("%d",&n);
  printf("Fibonacci Sequence:");
  for(i=1;i<=n;i++)
  {
   printf("%d ",first);
   temp=first+second;
  first=second;
  second=temp;
  }
  return 0;
 }
```

Output:

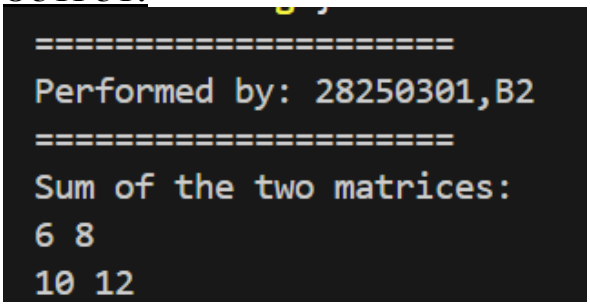
```
======================
Performed by: 28250301,B2
======================
Enter number of terms... 6
Fibonacci Sequence:0 1 1 2 3 5
```

Practical no.11

Aim: Write a program for addition of two matrices.
Software Required : Visual Studio Code
Code:

```
#include<stdio.h>
int main()
{
  printf("======================\n");
  printf("Performed by: 28250301,B2\n");
  printf("======================\n");

  int a[2][2]={{1,2},{3,4}};
  int b[2][2]={{5,6},{7,8}};
  int sum[2][2];
  int i,j;
  for(i=0;i<2;i++){
  for(j=0;j<2;j++){
  sum[i][j]=a[i][j]+b[i][j];
 }
}
  printf("Sum of the two matrices:\n");
  for(i=0;i<2;i++){
  for(j=0;j<2;j++){
  printf("%d ",sum[i][j]);
}
  printf("\n");
}
  return 0;
}
```

OUTPUT:



Practical No. 12

Aim: Write a program to calculate the following series without pow() function x - x3/3! +x5/5! - x7/7! ... xn/n!

Software Requirements: Visual Studio Code

Code:

```
#include<stdio.h>
int main()
{
  printf("======================\n");
  printf("Performed by: 28250301,B2\n");
  printf("======================\n");

   int n,x;
   double sum=0; //taking double as we need to handle decimal part.
   //Lets prompt the user to enter value of n
   printf("Enter the required number of terms in the series: ");
   scanf("%d",&n);

   //Lets prompt the user to enter value of x
   printf("Enter the value of x: ");
   scanf("%d",&x);

   printf("\n");

   for(int i=1,term_count=1;i<=n;i=i+2,term_count++) //outer loop for i=1,3,5,7,9,.....
   {
      int x_pow=1;
      for(int j=1;j<=i;j++)//inner loop to find x raise to power i
      {
         x_pow = x * x_pow; //calculating x raise to power
      }
      printf("For %d: x_power is: %d\n\n",i,x_pow);

      int fact = 1;
      for(int k=i;k>0;k--)//inner loop for calculating 1!,3!,5!,7!...
      {
         fact = fact * k;}
```

```
        printf("for i = %d fact = %d\n\n",i,fact);
       double result = (double)x_pow/fact;  //explicitly casting x_pow
       //condition statement to add and subtract the term in the final sum
       if(term_count%2==1)
          sum = sum + result;
       else
          sum = sum - result;
       // Print the intermediate sum after each term
       printf("After term %d: term_count = %d, result = %lf, sum = %lf\n\n", i, term_count,result,
sum);
    }
    printf("\nFinal sum of the terms is: %lf\n\n",sum);
}
```

Output:

```
======================
Performed by: 28250301,B2
======================
Enter the required number of terms in the series: 6
Enter the value of x: 2

For 1: x_power is: 2

for i = 1 fact = 1

After term 1: term_count = 1, result = 2.000000, sum = 2.000000

For 3: x_power is: 8

for i = 3 fact = 6

After term 3: term_count = 2, result = 1.333333, sum = 0.666667

For 5: x_power is: 32

for i = 5 fact = 120

After term 5: term_count = 3, result = 0.266667, sum = 0.933333


Final sum of the terms is: 0.933333
```

Practical no.13

Aim: Write a program for calculating transpose of a matrix.
Software required: Visual Studio Code
Code:

```
#include <stdio.h>
int main()
{
  printf("=====================\n");
  printf("Performed by: 28250301,B2\n");
  printf("=====================\n");
  int a[2][2], b[2][2];
  int i, j;
  printf("Enter the elements of the 2x2 matrix:\n");
  for(i = 0; i < 2; i++){
  for(j = 0; j < 2; j++){
  scanf("%d", &a[i][j]);
 }
}
  for(i = 0; i < 2; i++){
  for(j = 0; j < 2; j++){
  b[j][i] = a[i][j];
 }
}
  printf("\nThe transpose of the matrix is:\n");
  for(i = 0; i < 2; i++){
  for(j = 0; j < 2; j++){
  printf("%d ", b[i][j]);
}
  printf("\n");
}
  return 0;
}
```

OUTPUT:

```
=====================
Performed by: 28250301,B2
=====================
Enter the elements of the 2x2 matrix:
12
24
36
45

The transpose of the matrix is:
12 36
24 45
```

Practical No. 14

Aim: Write a program for matrix multiplication by checking compatibility.

Software Requirements: Visual Studio Code

Code:

```
#include<stdio.h>
int main()
{
  printf("======================\n");
  printf("Performed by: 28250301,B2\n");
  printf("======================\n");
  int a[2][2]={{1,2},{3,4}};
  int b[2][2]={{10,20},{30,40}};
  int mul[2][2];
  int i,j,k;
  for(i=0;i<2;i++)
  for(j=0;j<2;j++){
  mul[i][j]=0;
  for(k=0;k<2;k++)
  mul[i][j]+=a[i][k]*b[k][j];
}
  printf("Resultant Matrix:\n");
  for(i=0;i<2;i++){
  for(j=0;j<2;j++)
  printf("%d ",mul[i][j]);
  printf("\n");
}
  return 0;
}
```

OUTPUT:

```
======================
Performed by: 28250301,B2
======================
Resultant Matrix:
70 100
150 220
```

Practical No. 15

Aim: Write a program to concatenate two strings.

Software Requirements: Visual Studio Code

Code:

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

int main()
{
   printf("======================\n");
   printf("Performed by: 28250301, B2\n");
   printf("======================\n");

   char str1[50], str2[50];
   printf("Enter first string: ");
   scanf("%s", str1);

   printf("Enter second string: ");
   scanf("%s", str2);
   strcat(str1, str2);

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

OUTPUT:

```
======================
Performed by: 28250301, B2
======================
Enter first string: 12
Enter second string: 24
Concatenated string: 1224
```

Program.16

Aim: Write a program to implement Structure for storing information of a student.

Software Requirements: Visual Studio Code

Code:

```
#include <stdio.h>
struct Student
{
  char name[100];
  int rollNo;
  float marks;
};
   void displayStudentInfo(struct Student student) {
   printf("Student Information:\n");
   printf("---------------------------\n");
   printf("Student Name: %s\n", student.name);
   printf("Roll Number: %d\n", student.rollNo);
   printf("Marks: %.2f\n", student.marks);
   }
int main()
{
  struct Student student1;
  printf("=====================\n");
  printf("Performed by: 28250301, B2\n");
  printf("=====================\n");
  sprintf(student1.name, "AMRITA");
  student1.rollNo = 28250301;
  student1.marks = 99;
  displayStudentInfo(student1);
   return 0;
}
```

OUTPUT:

```
=====================
Performed by: 28250301, B2
=====================

Student Information:
---------------------------
Student Name: AMRITA
Roll Number : 28250301
Marks       : 99.00
```

Program.17

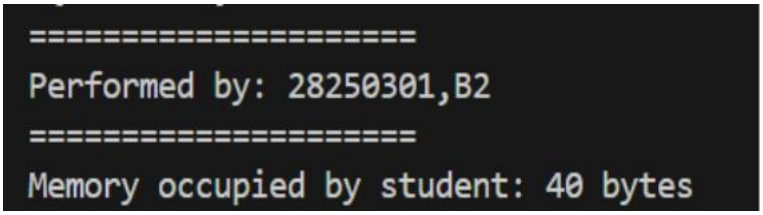
Aim: Write a program to implement Union.

Software Requirements: Visual Studio Code

Code:

```
#include <stdio.h>
union students
{
   char name[20];
   int rollno;
   float adm_no;
   char address[40];
};

int main()
{
   union students student1;
   printf("Performed by: 28250301, B2\n");
   printf("---------------------------\n");
   printf("Memory occupied by student: %d bytes\n", sizeof(student1));
   return 0;
}
```

OUTPUT:-



Program.18

Aim: Write a program to print the element of array using pointers.

Software Requirements: Visual Studio Code

Code:

```
#include <stdio.h>
int main() {
int arr[] = {11, 21, 31, 41, 51};
int *ptr;
ptr = arr;
printf("====================\n");
printf("Performed by: 28250301, B2\n");
printf("Name: AMRITA\n");
printf("====================\n");
printf("Array elements using pointers:\n");
for (int i = 0; i < 5; i++) {
printf("%d ", *(ptr + i));
}
printf("\n");
return 0;
}
```

OUTPUT:

```
====================
Performed by: 28250301, B2
Name: AMRITA
====================
Array elements using pointers:
11 21 31 41 51
```

Program.19

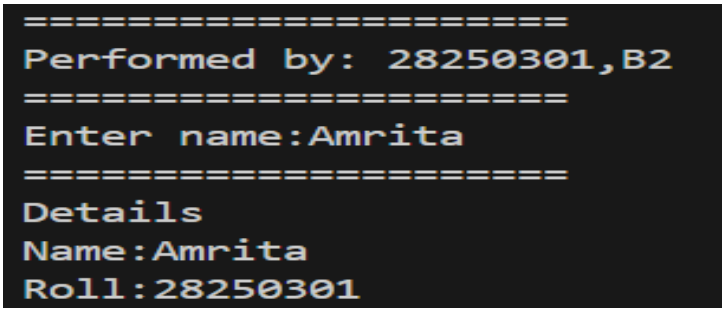
Aim: Write a program to print the elements of a structure using pointers.

Software Requirements: Visual Studio Code

Code:

```
#include <stdio.h>
struct Student{
char name[50];
int roll;
};
int main()
{
  struct Student s,*ptr;
  ptr=&s;
  printf("======================\n");
  printf("Performed by: 28250301,B2\n");
  printf("======================\n");
  printf("Enter name:");
  scanf("%s",ptr->name);
  ptr->roll=28250301;
  printf("======================\n");
  printf("Details\n");
  printf("Name:%s\n",ptr->name);
  printf("Roll:%d",ptr->roll);
  return 0;
}
```

OUTPUT:



Program.20

Aim: Write a program to explore malloc and calloc.

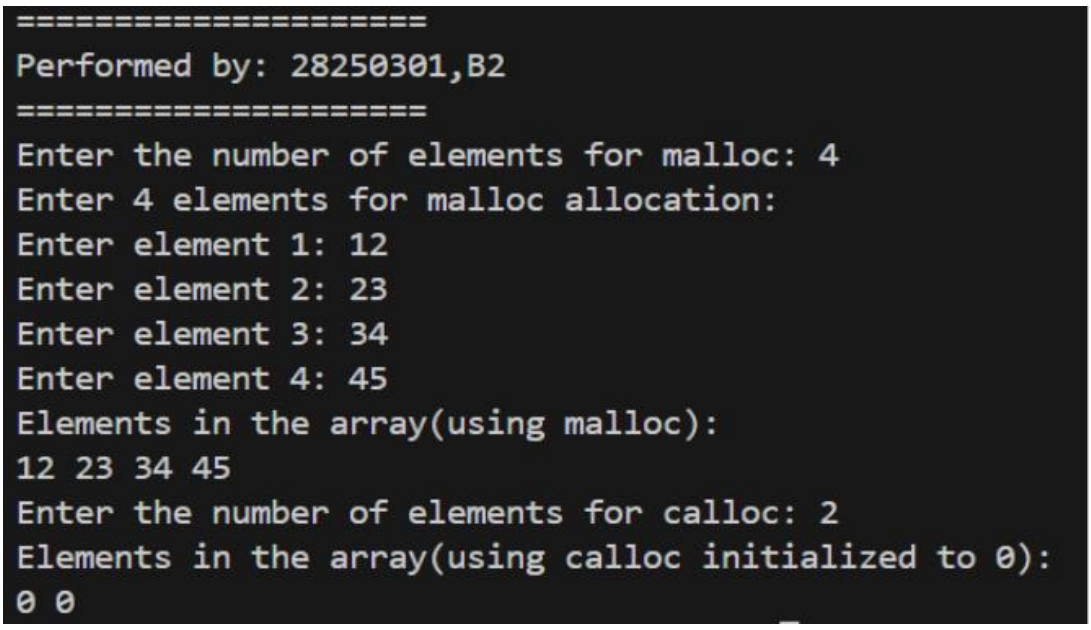
Software Requirements: Visual Studio Code

Code:

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
  int *arr_malloc, *arr_calloc;
  int size, i;
  printf("======================\n");
  printf("Performed by: 28250301, B2\n");
  printf("======================\n");

  printf("Enter the number of elements for malloc: ");
  scanf("%d", &size);
  arr_malloc = (int *)malloc(size * sizeof(int));

  if (arr_malloc == NULL)
{
   printf("Memory allocation failed using malloc.\n");
   return 1;
}
  printf("Enter %d elements for malloc allocation:\n", size);
  for (i = 0; i < size; i++)
{
  printf("Enter element %d: ", i + 1);
  scanf("%d", &arr_malloc[i]);
}

  printf("Elements in the array (using malloc):\n");
  for (i = 0; i < size; i++)
{
  printf("%d ", arr_malloc[i]);
}
  printf("\n");
  printf("Enter the number of elements for calloc: ");
```

```
  scanf("%d", &size);
arr_calloc = (int *)calloc(size, sizeof(int));

if (arr_calloc == NULL)
{
  printf("Memory allocation failed using calloc.\n");
  return 1;
}
  printf("Elements in the array (using calloc initialized to 0):\n");
  for (i = 0; i < size; i++)
{
  printf("%d ", arr_calloc[i]);
}
  printf("\n");
  free(arr_malloc);
  free(arr_calloc);
  return 0;
}
```

OUTPUT:



Practical No. 21

Aim: Write a program to create a file.

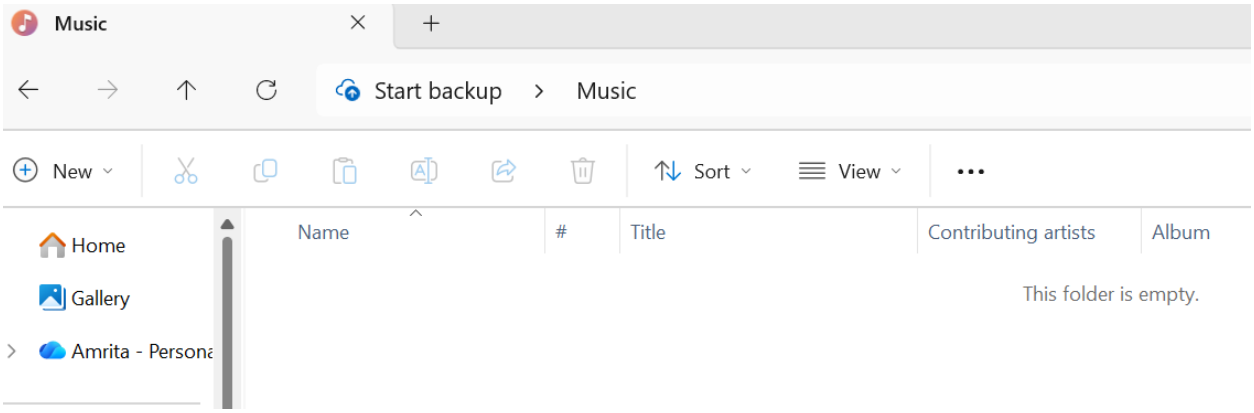
Software Requirements: Visual Studio Code

Code:

```
#include <stdio.h>
int main()
{
   FILE *fp;
   printf("======================\n");
   printf("Performed by: 28250301, B2\n");
   printf("======================\n");

   fp = fopen("C:\\Users\\codew\\Music\\file_handling.txt", "w");

   if (fp == NULL)
   {
      perror("File not created");
      return 1;
   }

   fprintf(fp, "Hello, this is file handling in C.");
   printf("File created successfully!");

   fclose(fp);
   return 0;
}
```

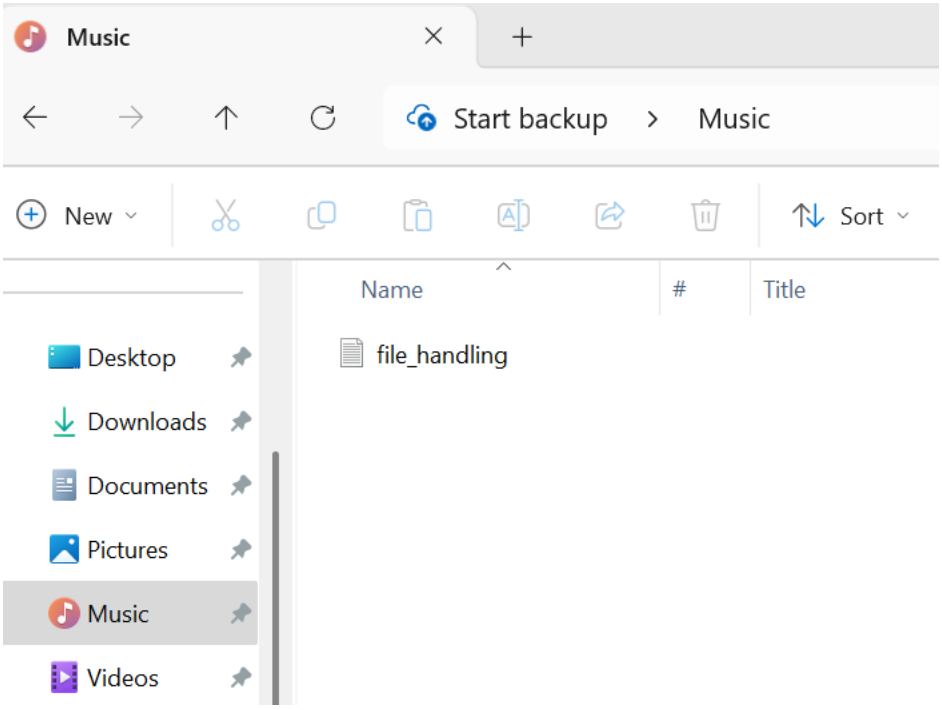
Output:

Before executing the code, the Music folder contains nothing:



After executing the code:

```
====================
Performed by: 28250301, B2
====================
File created successfully!
```



Practical No. 22

Aim: Write a program which copies one file to another.

Software Requirements: Visual Studio Code

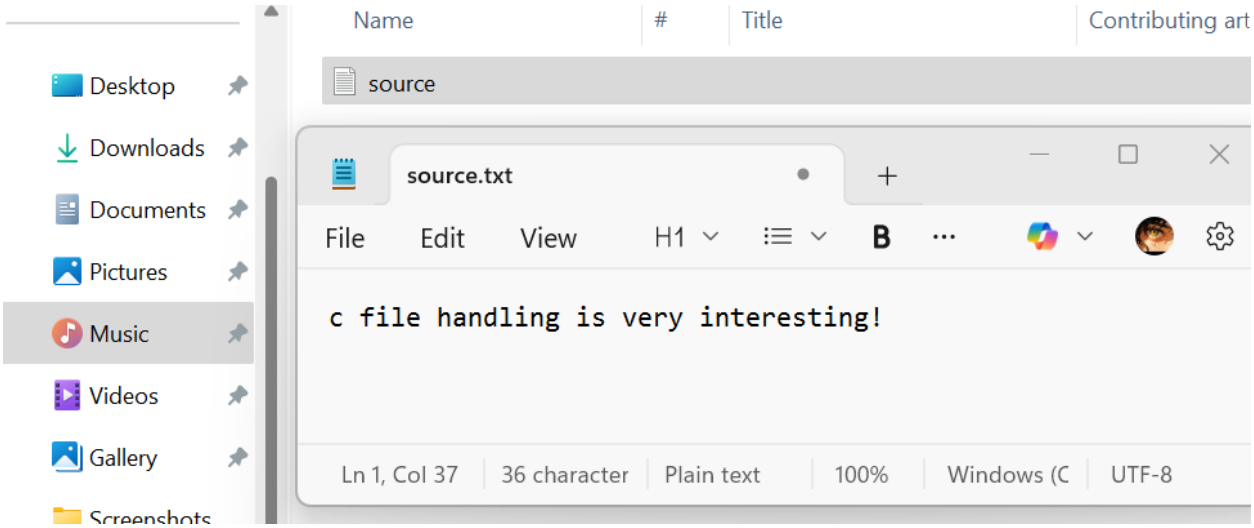
Code:

```
 #include <stdio.h>
int main()
{
  FILE *fs, *fd; int ch;
  printf("=====================\n");
  printf("Performed by: 28250301, B2\n");
  printf("=====================\n");
  /* Open source file in read mode */
  fs = fopen("C:\\Users\\codew\\Music\\source.txt", "r");
  if (fs == NULL)
  {
   perror("Source file not found");
   return 1;
  }

/* Open destination file in write mode */
fd = fopen("C:\\Users\\codew\\Music\\destination.txt", "w");
if (fd == NULL)
{
   perror("Destination file not created");
   fclose(fs);
   return 1;
}

/* Copy contents character by character */
while ((ch = fgetc(fs)) != EOF)
{
   fputc(ch, fd);
}

fclose(fs);
fclose(fd);
```

```
printf("File copied successfully!");
return 0;

}
```

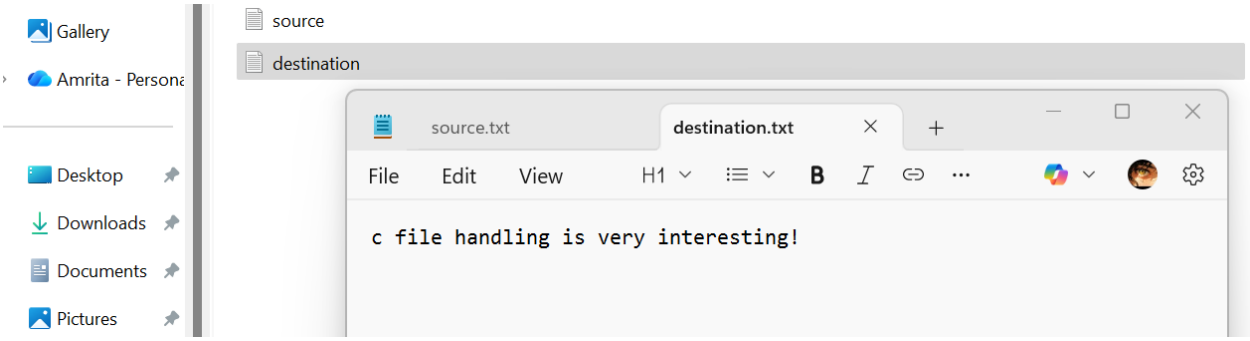
Output:

Source file contents:



Destination file contents after executing the code:

```
======================
Performed by: 28250301, B2
======================
File created successfully!
```



Practical No. 23

Aim: Write a program that counts the number of characters and number of lines in a text file.

Software Requirements: Visual Studio Code

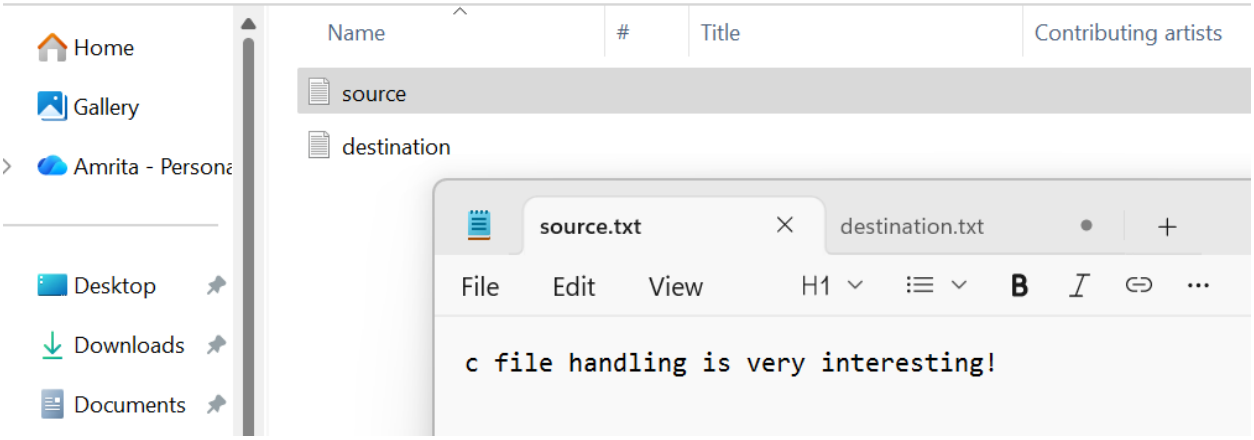
Code:

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

int main()
{
   FILE *fp;
   int ch;
   int nol = 0, not = 0, noc = 0, nob = 0;

   printf("=====================\n");
   printf("Performed by: 28250301, B2\n");
   printf("=====================\n");

   fp = fopen("C:\\Users\\codew\\Music\\source.txt", "r");

   if (fp == NULL)
   {
      printf("No such file found\n");
      exit(1);
   }

   while ((ch = fgetc(fp)) != EOF)
   {
      noc++;

      if (ch == ' ')
         nob++;

      if (ch == '\n')
         nol++;

      if (ch == '\t')
         not++;
```

```
    }

    fclose(fp);

    if (noc > 0)
        nol++;   // Count the last line

    printf("\nNumber of characters = %d\n", noc);
    printf("Number of blanks     = %d\n", nob);
    printf("Number of tabs       = %d\n", not);
    printf("Number of lines      = %d\n", nol);

    return 0;
}
```

Output:

Contents of source file:



Output after executing the code:

```
====================
Performed by: 28250301, B2
====================

Number of characters = 36
Number of blanks     = 5
Number of tabs       = 0
Number of lines      = 1
```

Practical No. 24

Aim: Write a program that changes every 5th character of data file into uppercase

Software Requirements: Visual Studio Code

Code:

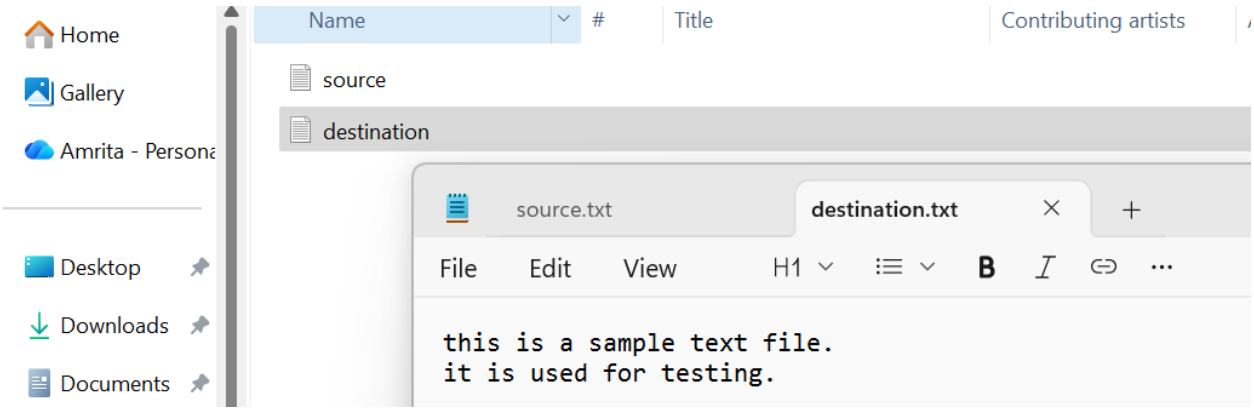
```
#include <stdio.h>
#include <ctype.h>   // for toupper()

int main()
{
   FILE *fp;
   char ch;
   int count = 0;

   printf("======================\n");
   printf("Performed by: 28250301, B2\n");
   printf("======================\n");

   // Open file in read + write mode
   fp = fopen("C:\\Users\\codew\\Music\\destination.txt", "r+");

   if (fp == NULL)
   {
      printf("Error opening file!\n");
      return 1;
   }

   // Loop through file character by character
   while ((ch = fgetc(fp)) != EOF)
   {
      count++;

      // Check if this is every 5th character
      if (count % 5 == 0)
      {
         // Convert to uppercase
         fseek(fp, -1, SEEK_CUR);   // move back one character
         fputc(toupper(ch), fp);
         fflush(fp);                // ensure data is written
      }
   }

   fclose(fp);

   printf("Every 5th character has been converted to uppercase successfully!\n");
```

```
    return 0;
}
```

Output:

Before executing the code, the content of file is:

```
====================
Performed by: 28250301, B2
====================
Every 5th character has been converted to uppercase successfully!
```



After executing the code: