



DAYANANDA SAGAR
UNIVERSITY



SCHOOL OF
COMPUTER APPLICATIONS

PROGRAMMING IN C LAB			
[As per the Choice Based Credit System (CBCS) scheme]			
SEMESTER -I			
Course Code	: 25CA1106	Credits	: 1
Hours / Week	: 2	Total Hours	: 30
L-T-P	: 0:0:2		
Course Learning Objectives: This Course will enable students to: <ol style="list-style-type: none"> 1. Develop the basic C programming approaches for analyzing the problem 2. To introduce C language concepts required for building a C program 3. Develop essential skills to build a pointer program. 4. Usage of different C concepts that support different sets of operations, which are suitable for various applications. 5. Implement dynamic memory allocation and C language concepts required for building real-world applications 			
List of Experiments			
1. Write a C program to find the sum and reverse of a number.			
2. Write a C program to find whether the given integer is a prime number or not			
3. Design a C program to implement the pyramid pattern using looping statements			
4. Design a program that takes three coefficients (a, b, and c) of a Quadratic equation ($ax^2+bx+c=0$) as input and compute all possible roots.			
5. Write a program to calculate the factorial of a given number.			
6. Write a program to calculate the nth Fibonacci number given the first two numbers in the series.			
7. Write a program to calculate the GCD of two numbers using recursion.			
8. Write a program to simulate a calculator with the following operations: Addition, Subtraction, Multiplication, Division – using functions, switch, and break			
9. String handling: <ol style="list-style-type: none"> a. Write a function to reverse the string in reverse and display it. (Strings) b. Write a function to concatenate the two strings without using strcat. (Strings) c. Write a function to find the length of the string. 			
10. Write a program to multiply two matrices using separate functions for input add matrices, display matrix and main function.			



1. Write a C program to find the sum and reverse of a number.

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

    int num, sum = 0, reverse = 0, digits, temp;

    printf("Enter a number: ");

    scanf("%d", &num);

    temp = num; // store original number

    while (temp > 0) {

        digits = temp % 10;

        sum += digits;

        reverse = reverse * 10 + digits;

        temp /= 10;

    }

    printf("Sum of digits = %d\n", sum);

    printf("Reverse of number = %d\n", reverse);

    return 0;

}
```

Output:

```
Enter a number: 123
Sum of digits = 6
Reverse of number = 321

Process returned 0 (0x0)    execution time : 4.020 s
Press any key to continue.
```



2. Write a C program to find whether the given integer is a prime number or not

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

    int num, i, isPrime = 1;
    printf("Enter an integer: ");

    scanf("%d", &num);

    if (num <= 1) {
        isPrime = 0;
    } else {
        for (i = 2; i <= num / 2; i++) {
            if (num % i == 0) {
                isPrime = 0;
                break;
            }
        }
    }

    if (isPrime == 1)

        printf("%d is a Prime number.\n", num);
    else

        printf("%d is Not a Prime number.\n", num);

    return 0;
}
```



3. Design a C program to implement the pyramid pattern using looping statements

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

    int i, j, rows, space;
    printf("Enter number of rows: ");

    scanf("%d", &rows);

    for (i = 1; i <= rows; i++) {

        for (space = 1; space <= rows - i; space++) {

            printf(" ");

        }
        for (j = 1; j <= (2 * i - 1); j++) {

            printf("*");

        }

        printf("\n"); // move to next line

    }

    return 0;

}
```

```
Enter number of rows: 5
```

```
  *
 ***
*****
*****
*****
```

```
Process returned 0 (0x0)   execution time : 3.851 s
Press any key to continue.
```



4. Design a program that takes three coefficients (a, b, and c) of a Quadratic equation ($ax^2+bx+c=0$) as input and compute all possible roots.

```
#include <stdio.h>
#include <math.h>
int main() {
    int a,b,c,d;
    float root1,root2;
    printf("Enter the coefficients a, b and c: ");
    scanf("%d %d %d",&a,&b,&c);
    d = b*b - 4*a*c; // calculating the discriminant
    if(d<0){
        printf("both roots are complex and imaginary\n");
    }
    if(d==0){
        printf("both roots are real and equal\n");
        float root = -b/(2.0*a);
        printf("the root is: %.2f\n",root);
    }
    if(d>0){
        printf("both roots are real and distinct\n");
        root1 = (-b + sqrt(d))/(2.0*a);
        root2 = (-b - sqrt(d))/(2.0*a);
        printf("the roots are: %.2f and %.2f\n",root1,root2);
    }
    return 0;
}
```

```
Enter the coefficients a, b and c: 1 7 6
both roots are real and distinct
the roots are: -1.00 and -6.00

Process returned 0 (0x0)    execution time : 9.000 s
Press any key to continue.
```



5. Write a program to calculate the factorial of a given number.

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

    int n, i;

    int long long fact = 1;
    printf("Enter a positive integer: ");

    scanf("%d", &n);

    if (n < 0) {

        printf("Factorial of a negative number doesn't exist.\n");

    } else {

        for (i = 1; i <= n; i++) {

            fact *= i;

        }

        printf("Factorial of %d = %d\n", n, fact);

    }

    return 0;

}
```

Output:

```
Enter a positive integer: 5
Factorial of 5 = 120

Process returned 0 (0x0)   execution time : 3.275 s
Press any key to continue.
```

6. Write a program to calculate the nth Fibonacci number given the first two numbers in the series.

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

```
int main(){
```

```
    int n, a = 0, b = 1, c;
```

```
    printf("Enter the number of terms you want in Fibonacci series\n");
```

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

```
    printf("Fibonacci series:\n");
```

```
    for ( int i = 0 ; i < n ; i++ )
```

```
    {
```

```
        printf("%d\n",a);
```

```
        c = a + b;
```

```
        a = b;
```

```
        b = c;
```

```
    }
```

```
    return 0;
```

```
}
```

```
Enter the number of terms you want in Fibonacci series
```

```
10
```

```
Fibonacci series:
```

```
0
```

```
1
```

```
1
```

```
2
```

```
3
```

```
5
```

```
8
```

```
13
```

```
21
```

```
34
```

```
Process returned 0 (0x0)    execution time : 16.154 s
```

```
Press any key to continue.
```



7. Write a program to calculate the GCD of two numbers using recursion.

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

```
int gcd(int a, int b) {
```

```
    if (b == 0)
```

```
        return a;
```

```
    else
```

```
        return gcd(b, a % b);
```

```
}
```

```
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));
```

```
    return 0;
```

```
}
```




8. Write a program to simulate a calculator with the following operations: Addition, Subtraction, Multiplication, Division – using functions, switch, and break

```
#include <stdio.h>
float add(float a, float b) {
    return a + b;
}
float subtract(float a, float b) {
    return a - b;
}
float multiply(float a, float b) {
    return a * b;
}
float divide(float a, float b) {
    if (b != 0)
        return a / b;
    else {
        printf("Error! Division by zero is not allowed.\n");
        return 0;
    }
}
```



DAYANANDA SAGAR
UNIVERSITY



SCHOOL OF
COMPUTER APPLICATIONS

```
int main() {  
  
    int choice;  
  
    float num1, num2, result;  
  
  
    printf("Select an operation to perform:\n");  
    printf("1. Addition\n");  
    printf("2. Subtraction\n");  
    printf("3. Multiplication\n");  
    printf("4. Division\n");  
    printf("Enter your choice (1-4): ");  
    scanf("%d", &choice);  
  
  
    printf("Enter two numbers: ");  
    scanf("%f%f", &num1, &num2);  
  
  
    switch (choice) {  
  
        case 1:  
            result = add(num1, num2);  
            printf("Result = %.2f\n", result);  
            break;  
  
        case 2:  
            result = subtract(num1, num2);  
            printf("Result = %.2f\n", result);  
            break;  
  
        case 3:  
            result = multiply(num1, num2);  
            printf("Result = %.2f\n", result);  
            break;
```



case 4:

```
result = divide(num1, num2);  
if (num2 != 0)  
    printf("Result = %.2f\n", result);  
break;
```

default:

```
printf("Invalid choice!\n");
```

```
}
```

```
return 0;
```

```
}
```

```
C:\Users\adilm\OneDrive\Des  X  +  v  
Select an operation to perform:  
1. Addition  
2. Subtraction  
3. Multiplication  
4. Division  
Enter your choice (1-4): 3  
Enter two numbers: 10  
5  
Result = 50.00  
  
Process returned 0 (0x0)   execution time : 12.283  
Press any key to continue.
```

9. String handling:

- d. Write a function to reverse the string in reverse and display it. (Strings)
- e. Write a function to concatenate the two strings without using strcat. (Strings)
- f. Write a function to find the length of the string.



DAYANANDA SAGAR
UNIVERSITY



SCHOOL OF
COMPUTER APPLICATIONS

```
#include <stdio.h>

#include <string.h>

int length(char str[]){
    int length=0;
    int count=0;
    while(str[length]!='\0'){
        count++;
        length++;
    }
    return count;
}

void concat(char str1[],
char str2[]){
    int i=0,j=0;
    char result[200];
    while(str1[i]!='\0'){
        result[j++]=str1[i++];
    }
    i=0;
    while(str2[i]!='\0'){
        result[j++]=str2[i++];
    }
    result[j]='\0';
    printf("Concatenated
string: %s\n",result);
}

void reverse(char str[]){
    int len=strlen(str);
    char rev[100];
    int i, j;
```

```
for(i=len-1,j=0;i>=0;i--
,j++){
    rev[j]=str[i];
}
rev[j]='\0';
printf("Reversed
String:%s\n",rev);
}
#include <stdio.h>

//int length(char[]);

int main()
{
char str1[100],str2[100];

printf("Enter first
string\n");

gets(str1);

printf("Enter second
string\n");

gets(str2);

printf("\n-- String
operations--\n");

printf("Length of a
string is
%d\n",length(str1));

reverse(str1);

concat(str1,str2);

return 0;
}
```

Output:

C:\Users\adilm\OneDrive\Des ×

+

▼

Enter first string

hello

Enter second string

world

-- String operations--

Length of a string is 5

Reversed String:olleh

Concatenated string: helloworld

Process returned 0 (0x0) execution time : 9.169 s

Press any key to continue.

|



Q10 Write a program to multiply two matrices using separate functions for input add matrices, display matrix and main function.

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

```
void input_matrix(int mat[10][10], int rows,int cols){
    int i,j;
    printf("Enter elements of the matrix (%d X %d):\n",rows,cols);
    for(i=0;i<rows;i++){
        for(j=0;j<cols;j++){
            scanf("%d",&mat[i][j]);
        }
    }
}
```

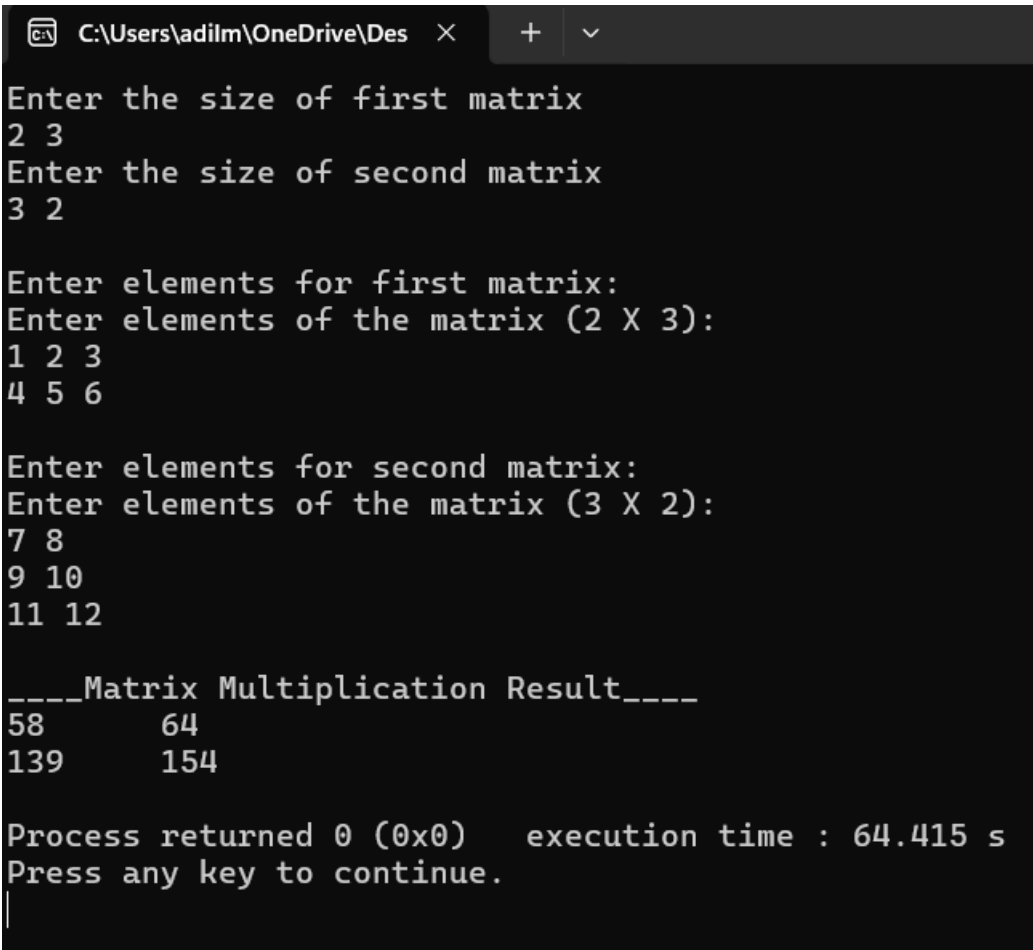
```
void multiply_Matrix(int A[10][10], int B[10][10],int C[10][10],int r1 ,int c1,int c2){
    int i,j,k;
    //initialize result matrix C to 0
    for(int i=0;i<r1;i++){
        for(j=0;j<c2;j++){
            C[i][j]=0;
            for(k=0;k<c1;k++){
                C[i][j]+=A[i][k]*B[k][j];
            }
        }
    }
}
```

```
void display_Matrix(int mat[10][10],int row,int col){
    int i,j;
    for(i=0;i<row;i++){
        for(j=0;j<col;j++){
            printf("%d\t",mat[i][j]);
        }
        printf("\n");
    }
}
```

```

#include <stdio.h>
int main()
{
int A[10][10],B[10][10],C[10][10];
int r1,c1,r2,c2;
printf("Enter the size of first matrix\n");
scanf("%d%d",&r1,&c1);
printf("Enter the size of second matrix\n");
scanf("%d%d",&r2,&c2);
if(c1!=r2){
    printf("matrix multiplication not possible\n");
    return 0;
}
printf("\nEnter elements for first matrix:\n");
input_matrix(A,r1,c1);
printf("\nEnter elements for second matrix:\n");
input_matrix(B,r2,c2);
multiply_Matrix(A,B,C, r1,c1,c2);
printf("\n____Matrix Multiplication Result____\n");
display_Matrix(C,r1,c2);
return 0;
}

```



```

C:\Users\adilm\OneDrive\Des  ×  +  ▾
Enter the size of first matrix
2 3
Enter the size of second matrix
3 2

Enter elements for first matrix:
Enter elements of the matrix (2 X 3):
1 2 3
4 5 6

Enter elements for second matrix:
Enter elements of the matrix (3 X 2):
7 8
9 10
11 12

____Matrix Multiplication Result____
58      64
139     154

Process returned 0 (0x0)   execution time : 64.415 s
Press any key to continue.
|

```