

1. Senior Practical Physical Chemistry, B.D. Khosla, A. Gulati and V.Garg (R. Chand & Co., Delhi)
2. An Introduction to Practical Chemistry, K. K. Sharma and D.S. Sharma (Vikas publishing, N. Delhi)

2. An Introduction to Practical Chemistry, K. K. Sharma and D.S. Sharma (Vikas publishing, N. Delhi)							
Course Code	Course Title						Core / Elective
ES 155 CS	Programming for Problem Solving Lab (Common to All Branches)						Core
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	4	25	50	2

Course Objectives

- Understand the fundamentals of programming in C Language.
- Write, compile and debug programs in C.
- Formulate solution to problems and implement in C.
- Effectively choose programming components to solve computing problems

Course Outcomes

The students will be able to

- Choose appropriate data type for implementing programs in C language.
- Design and implement modular programs involving input output operations, decision making and looping constructs.
- Implement search and sort operations on arrays.
- Apply the concept of pointers for implementing programs on dynamic memory management and string handling.
- Design and implement programs to store data in structures and files.

Programming Exercise:

1. Finding maximum and minimum of given set of numbers, finding roots of quadratic equation.
2. Sin x and Cos x values using series expansion.
3. Conversion of binary to decimal, octal, hexadecimal and vice versa.
4. Generating Pascal triangle, pyramid of numbers.
5. Recursion: factorial, Fibonacci, GCD.
6. Matrix addition and multiplication using arrays, linear search and binary search using recursive and non-recursive procedures.
7. Bubble sort and selection sort.
8. Programs on pointers: pointer to arrays, pointer to functions.
9. Functions for string manipulations.
10. Programs on structures and unions.
11. Finding the number of characters, words and lines of given text file.
12. File handling programs

Suggested Readings:

1. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill
2. A.K. Sharma, Computer Fundamentals and Programming in C, Universities Press, 2018.
3. E. Balaguruswamy, Programming in ANSI C, Tata McGraw-Hill
4. Brian W. Kernighan and Dennis M. Ritchie, the C Programming Language, Prentice Hall of India.

Program 1(a):-

* C program to find maximum and minimum element in array */

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

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

```
#define MAX_SIZE 100
```

```
int main()
```

```
{
```

```
int arr[MAX_SIZE];
```

```
int i, max, min, size;
```

```
printf("Enter size of an array:");  
scanf("%d", &size);
```

```
printf("Enter elements in the array:");  
for(i=0; i<size; i++)
```

```
{  
scanf("%d", &arr[i]);
```

```
}  
max = arr[0];  
min = arr[0];
```

```
for(i=1; i<size; i++)
```

```
{  
if(arr[i] > max)
```

```
{
```

17 Saturday
FEBRUARY

```
max = arr[i];
```

```
{ if (arr[i] < min)
```

```
{ min = arr[i];
```

```
}
```

```
printf("Maximum element = %d \n", max);
```

```
printf("Minimum element = %d \n", min);
```

```
return 0;
```

```
}
```

Output :-

18 Sunday

04/3/18 Week 7

MARCH 2018						
M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	-	-	-	-

16 Program 5:- Write a C program to find Quadratic Equation of roots

```

10 #include <math.h>
11 #include <stdio.h>
12 void main()
13 {
14     int a, b, c;
15     float r1, r2;
16     printf("Enter values of a, b, c: \n");
17     scanf("%d %d %d", &a, &b, &c);
18
19     if (b*b > 4*a*c)
20     {
21         r1 = (-b + sqrt(b*b - 4*a*c)) / (2*a);
22         r2 = (-b - sqrt(b*b - 4*a*c)) / (2*a);
23
24         printf("The values of root of Equation are");
25         printf(" r1 = %f \t r2 = %f", r1, r2);
26     }
27     else
28         printf("\n Roots are imaginary");
29     getch();

```


17 Saturday
MARCH

076-289 • Week 11

2(a)

/* C program to calculate the value of $\sin(x)$ */

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

```
int main()
```

```
{
```

```
    int i, j, n, fact, sign = -1;
```

```
    float x, p, sum = 0;
```

```
    printf("Enter the value of x:");
```

```
    scanf("%f", &x);
```

```
    printf("Enter the value of n:");
```

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

```
    for(i=1; i<=n; i+=2)
```

```
    {
```

```
        p=1;
```

```
        fact=1;
```

```
        for(j=1; j<=i; j++)
```

```
        {
```

```
            p=p*x;
```

```
            fact=fact*j;
```

```
        }
```

18 Sunday

077-288 Week 11

APRIL 2018
W T F S S

09 $\text{Sign} = -1 * \text{sign};$

10 $\text{Sum} += \text{sign} * p / \text{fact};$

11 $\text{printf}(" \sin \%0.2f = \%0f", x, \text{sum});$

12 $\text{return } 0;$

13 $\}$

```
/* program to calculate the value of  $\cos(x)$  */
```

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

```
int main()
```

```
{  
    int i, j, n, fact, sign = -1;  
    float x, p, sum = 0;
```

```
    printf("Enter the value of x: ");
```

```
    scanf("%f", &x);
```

```
    printf("Enter the value of n: ");
```

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

```
    for(i = 2; i <= n; i += 2)
```

```
    {  
        p = 1;
```

```
        fact = 1;
```

```
        for(j = 1; j <= i; j++)
```

```
        {  
            p = p * x;
```

```
            fact = fact * j;
```

APRIL 2018						
W	M	T	W	T	F	S
13	30					1
14	2	3	4	5	6	7
15	9	10	11	12	13	14
16	16	17	18	19	20	21
17	23	24	25	26	27	28
						29

}

sum += sign * p / fact;

sign = -1 * sign;

}

printf("cos %0.2f = %f", x, 1 + sum);

return 0;

}

3(a)

/* program to convert binary number to decimal */

#include <stdio.h>

#include <math.h>

int convertBinaryToDecimal(long long n);

int main()

{

long long n;

printf("Enter a binary number:");

scanf("%lld", &n);

printf("%lld in binary = %d in decimal", n,
convertBinaryToDecimal(n));

return 0;

}

int convertBinaryToDecimal(long long n)

{

int decimalNumber = 0, i = 0, remainder;

while(n != 0)

```
08  
09 { remainder = n % 10;  
10   n /= 10;
```

```
11 decimalNumber += remainder * pow(2, i);  
12   ++i;
```

```
13 } return decimalNumber;
```

```
14 }
```


24 Saturday
MARCH

083-282 Week 12

2(b)

/* program to convert Binary to Octal */

#include <stdio.h>

#include <math.h>

int ConvertBinarytoOctal(long long binaryNumber);
int main()

{ long long binaryNumber;

printf("Enter a binary number");

scanf("%lld", &binaryNumber);

printf("%lld in binary = %d in octal", binaryNumber,
ConvertBinarytoOctal(binaryNumber));

return 0;

}

25 Sunday

084-281 Week 12

int ConvertBinarytoOctal(long long binaryNumber)

{

int octalNumber = 0, decimalNumber = 0, i = 0;

while (binaryNumber != 0)

{

APRIL 2018						
W	T	W	T	F	S	S
13	30					1
14	2	3	4	5	6	7
15	9	10	11	12	13	14
16	16	17	18	19	20	21
17	23	24	25	26	27	28

2018

```

decimalNumber += (binaryNumber % 10)
                * pow(2, i);

```

```

++i;

```

```

binaryNumber /= 10;

```

```

}

```

```

i = 1;

```

```

while (decimalNumber != 0)

```

```

{

```

```

    OctalNumber += (decimalNumber % 8) * i;

```

```

    decimalNumber /= 8;

```

```

    i *= 10;

```

```

}

```

```

return OctalNumber;

```

```

}

```


3(c)-

/* Binary to Hexadecimal */

#include <stdio.h>

#include <conio.h>

void main()

{
long int binary-number, hexadecimal-number=0,
i=1, remainder;
clrscr();

printf("Enter any Binary Number");

scanf("%d", &binary-number);

while(binary-number != 0)

{
remainder = binary-number % 10;

hexadecimal-number = hexadecimal-number +
remainder * i;

i = i * 2;

binary-number = binary-number / 10;

}
printf("Equivalent Hexadecimal Number %lx",
hexadecimal-number);

getch();
}

APRIL							2018
W	M	T	W	T	F	S	S
13	30						1
14	2	3	4	5	6	7	8
15	9	10	11	12	13	14	15
16	16	17	18	19	20	21	22
17	23	24	25	26	27	28	29

27 Tuesday
FEBRUARY

058-307 Week 9

A(a)

/* Pascal Triangle */

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

```
int main()
```

```
{  
    int rows, col=1, space, i, j;
```

```
    printf("Enter number of rows");
```

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

```
    for(i=0; i<rows; i++)
```

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

```
        for(j=0; j<=i; j++)
```

```
        {
```

```
            if(j==0 || i==0)
```

```
                col=1;
```

```
            else
```

MARCH 2018						
W	M	T	W	T	F	S
9			1	2	3	4
10	5	6	7	8	9	10
11	12	13	14	15	16	17
12	19	20	21	22	23	24
13	26	27	28	29	30	31

```
cal = cal * (i-j+1)/j;
```

```
printf("%4d", cal);
```

```
}
```

```
printf("\n");
```

```
}
```

```
return 0;
```

```
}
```

Q6)

/* Write a 'C' program to print pyramid of numbers */

#include <stdio.h>

#include <conio.h>

void main()

{

int i, j, rows;

clrscr();

printf("Enter numbers of rows: ");

scanf("%d", &rows);

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

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

{ printf("%d", j);

}

printf("\n");

}

getch();

}

08 Program 5:-

09 a) // Recursion to calculate factorial //

10 #include <stdio.h>

#include <conio.h>

11 int fact(int);

void main()

12 {
int x, n;

13 clrscr();

printf("Enter the number to find factorial:");

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

16 x = fact(n);

17 printf("Factorial of %d is %d", n, x);

18 getch();

} int fact(int n)

{ if (n == 0)

return(1);

return (n * fact(n-1));

}

b) // Fibonacci of a Number Using Recursion

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

```
int Fibonacci(int);
```

```
int main()
```

```
{  
    int n, i = 0, c;
```

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

```
    printf("Fibonacci Series \n");
```

```
    for(c = 1; c <= n; c++)
```

```
{  
        printf("%d \n", Fibonacci(i));
```

```
        i++;
```

```
    }  
    return 0;
```

```
}
```

```
int Fibonacci(int n)
```

```
{  
    if (n == 0)
```

```
        return 0;
```

```
    else
```

```
        if (n == 1)
```

```
            return 1;
```

```
    else
```

```
        return (Fibonacci(n-1) + Fibonacci(n-2));
```

```
}
```


c) // Recursion of GCD Numbers.

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

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

```
int gcd(int, int);
```

```
void main()
```

```
{ int a, b, c;
```

```
printf("Enter the two numbers to find their  
gcd: ");
```

```
scanf("%d %d", &a, &b);
```

```
c = gcd(a, b);
```

```
printf("The gcd of %d and %d is %d\n",  
a, b, c);
```

1 APRIL
Sunday

091-274 Week 13

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

```
{ while(a != b)
```

```
{
```

APRIL 2018						
S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

2018

2 Monday
APRIL

Week 14 092-273

if (a > b)

{ return gcd(a-b, b);

}

else

{

return gcd(a, b-a);

}

}

return a;

}

1 Thursday
MARCH

6(a)3)

/ * Addition of two Matrices */

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

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

```
#include <math.h>
```

```
void main()
```

```
{
```

```
int a[3][3], b[3][3], c[3][3], i, j;
```

```
clrscr();
```

```
printf("\n Enter the number in Matrix A");
```

```
for(i=0; i<3; i++)
```

```
{
```

```
for(j=0; j<3; j++)
```

```
{
```

```
scanf("%d", &a[i][j]);
```

```
}
```

```
printf("\n Enter the number in Matrix B");
```

```
for(i=0; i<3; i++)
```

```
{
```

```
for(j=0; j<3; j++)
```

```
{
```

```
scanf("%d", &b[i][j]);
```

```
}
```



```
printf("\n Addition of two matrices c:");  
for(i=0; i<3; i++)  
{  
    for(j=0; j<3; j++)  
    {  
        c[i][j] = a[i][j] + b[i][j];  
    }  
}  
printf("\n The Resultant Matrix ");  
for(i=0; i<3; i++)  
{  
    for(j=0; j<3; j++)  
    {  
        printf("%2d", c[i][j]);  
    }  
    printf("\n");  
}  
getch();  
}
```

3 Saturday
MARCH

6(2)(ii)

/ * Multiplication Of Matrix * /

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
void main()
{
    int a[10][10], b[10][10], c[10][10], i, j, m, n, p, q, k;

    printf("\n Enter order of Matrix A:");
    scanf("%d %d", &m, &n);

    printf("\n Enter order of Matrix B");
    scanf("%d %d", &p, &q);

    if (m == q)
    {
        printf("Enter Elements in Matrix A:");
        for (i = 0; i < m; i++)
        {
            for (j = 0; j < n; j++)
            {
                scanf("%d", &a[i][j]);
            }
        }
    }
}
```

4 Sunday

063-302 Week 9

APRIL	M	T	W	T	F	S
13	1	2	3	4	5	6
14	7	8	9	10	11	12
15	13	14	15	16	17	18
16	19	20	21	22	23	24
17	25	26	27	28	29	30

printf("Enter elements in Matrix B");

for(i=0; i<=p; i++)

{ for(j=0; j<=q; j++)

{ scanf("%d", &b[i][j]);

}}

printf("Matrix Multiplication");

for(i=0; i<=m; i++)

{ for(j=0; j<=q; j++)

{

c[i][j]=0;

for(k=0; k<=n; k++)

{

c[i][j] = c[i][j] + (a[i][k] * b[k][j]);

}}

MARCH							201
W	M	T	W	T	F	S	
9	1	2	3	4	5	6	
10	5	6	7	8	9	10	1
11	12	13	14	15	16	17	1
12	19	20	21	22	23	24	2
13	26	27	28	29	30	31	

6 Tuesday
MARCH

065-700 Week 10

```
printf("\n The Resultant Matrix is: ");
```

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

```
{  
    for(j=0; j<=p; j++)
```

```
{
```

```
    printf("%5d", c[i][j]);
```

```
}
```

```
    printf("\n");
```

```
}  
}
```

```
else
```

```
printf("\n Matrix Cannot be Multiplied");
```

```
}
```

```
getch();
```

```
}
```

// C program for Linear Search Using Recursive Method.

#include <stdio.h>

int Search(int[], int, int);

int main()

{

int size, index, Key, list[20], count=0, i;
clrscr();

printf("Enter the size of the list: ");

scanf("%d", &size);

index = size;

printf("Printing the list: \n");

for(i=0; i<size; i++)

{

list[i] = rand() % size;

printf("%d \t", list[i]);

}

APRIL						
W	M	T	W	T	F	S
13	30					
14	2	3	4	5	6	
15	9	10	11	12	13	
16	16	17	18	19	20	
17	23	24	25	26	27	


```
printf("\n Enter the key to search:");  
scanf("%d", &key);  
while(index > 0)  
{  
    index = search(list, index-1, key);  
    printf("Key found at position: %d \n", index+1);  
    count++;  
}  
if(!count)  
    printf("Key not found. \n");  
return 0;  
}
```

```
int search(int array[], int size, int key)  
{  
    int location;
```

```
    if(array[size] == key)
```

```
    {  
        return size;  
    }
```

MAY						2018
W	M	T	W	T	F	S
18		1	2	3	4	5
19	7	8	9	10	11	12
20	14	15	16	17	18	19
21	21	22	23	24	25	26
22	28	29	30	31	-	-

else

if (size == -1)

{

return -1;

}

else

{

return (location = search(array, size-1, key));

}

}

APRIL							2018
W	M	T	W	T	F	S	S
13	30						1
14	2	3	4	5	6	7	8
15	9	10	11	12	13	14	15
16	16	17	18	19	20	21	22
17	23	24	25	26	27	28	29

// C program Linear Search non-recursive method //

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

int main()

{
int key, n, i;
clrscr();

printf("\n Enter the size of the array: ");

scanf("%d", &n);

printf("\n Enter %d integers: ", n);

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

scanf("%d", &a[i]);

printf("\n Enter a value to be searched: ");

scanf("%d", &key);

10 Tuesday
APRIL

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

```
if(Key == a[i])
```

```
{
```

```
printf("\n Key value is found at position \n", i+1);
```

```
exit(0);
```

```
}
```

```
printf("\n Given Key is not found in the list \n");
```

```
return 0;
```

```
}
```


//C program for binary search Using Recursion //

#include <stdio.h>

#include <stdlib.h>

#define size 10

int binsearch(int[], int, int, int);

int main()

{ int num, i, key, position, low, high, list[size];
clrscr();

printf("\n Enter the total number of elements ");

scanf("%d", &num);

printf("\n Enter the elements of list: ");

for(i=0; i<num; i++)

{ scanf("%d", &list[i]);

}

low = 0;
high = num - 1;

MAY 2018						
M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	-	-	-

08

09

10

11

12

13

15

16

17

18

```
printf("\n Enter element to be searched: ");
```

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

```
position = binsearch(list, key, low, high);
```

```
if (position != -1)
```

```
{
```

```
printf("\n Number present at %d", (position+1));
```

```
} else
```

```
printf("\n The number is not present in the list");
```

```
return 0;
```

```
}
```

```
int binsearch(int a[], int x, int low, int high)
```

```
{ int mid;
```

```
if (low > high)
```

```
return -1;
```

7 Saturday
APRIL

097-268 Week 14

mid = (low + high) / 2;

if (x == a[mid])

{ return (mid);

}

else

if (x < a[mid])

{ binsearch(a, x, low, mid - 1);

}

else

{

binsearch(a, x, mid + 1, high);

}

}

8 Sunday

098-267 V

MAY		2018						
W	M	T	W	T	F	S	S	
18		1	2	3	4	5	6	
19	7	8	9	10	11	12	13	
20	14	15	16	17	18	19	20	
21	21	22	23	24	25	26	27	
22	28	29	30	31	-	-	-	

C program code for binary search (Non-Recursive)

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

```
int main()
```

```
{  
    int c, first, last, middle, n, search, array[100];
```

```
    printf("Enter number of elements: \n");
```

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

```
    printf("Enter %d integer: \n", n);
```

```
    for(c=0; c<n; c++)
```

```
        scanf("%d", &array[c]);
```

```
    printf("Enter the value to find: \n");
```

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

```
    first = 0;
```

```
    last = n-1;
```

MAY						
	M	T	W	T	F	S
18		1	2	3	4	5
19	7	8	9	10	11	12
20	14	15	16	17	18	19
21	21	22	23	24	25	26
22	28	29	30	31	-	-

middle = (first + last) / 2;

while (first <= last)
{

if (array[middle] < search)

first = middle + 1;

else

if (array[middle] == search)

{

printf("%d is present at index %d \n", search,
middle + 1);

break;

}

else

last = middle - 1;

middle = (first + last) / 2;

}

if (first > last)

printf("Not found! %d is not present in the
list. \n", search);

return 0;

}

APRIL 2018						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Program 7: // C program Bubble Sort //

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

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

```
void main()
```

```
{  
    int a[20], i, j, temp, n;  
    clrscr();
```

```
    printf("Enter the size of the array \n");
```

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

```
    printf("Enter %d elements \n", n);
```

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

```
        scanf("%d", &a[i]);
```

```
    for(j=n-2; j>=0; j--)
```

```
        for(i=0; i<=j; i++)
```

```
            if(a[i] > a[i+1])
```

15 Sunday

MAY 2018						
W	M	T	W	T	F	S
18	1	2	3	4	5	6
19	7	8	9	10	11	12
20	14	15	16	17	18	19
21	21	22	23	24	25	26
22	28	29	30	31	-	-

```
{
    temp = a[i];
```

```
    a[i] = a[i+1];
```

```
    a[i+1] = temp;
```

```
}
```

```
printf("\n Sorting elements using bubble sort  
are: \n");
```

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

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

```
    getch();
```

```
}
```


// C program for selection sort //

#include <stdio.h>

#include <conio.h>

void main()

{
int a[50], i, j, n, temp;
clrscr();

printf("Enter the size of the array \n");

scanf("%d", &n);

printf("Enter %d elements \n", n);

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

scanf("%d", &a[i]);

for(i=0; i<=n-2; i++)

for(j=i+1; j<=n-1; j++)

if(a[i] > a[j])

{

MAY 2018						
W	M	T	W	T	F	S
18		1	2	3	4	5
19	7	8	9	10	11	12
20	14	15	16	17	18	19
21	21	22	23	24	25	26
22	28	29	30	31	-	-

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

printf("Elements in sorted order: \n");

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

printf("%d \n", a[i]);

getch();

}

Program 8: Programs On pointers

// C program for pointer to array //

#include <stdio.h>

void main()

{

int i, arr[5];

int *ptr;

ptr = arr;

printf("Enter 5 Elements");

for(i=0; i<5; i++)

{

scanf("%d", ptr);

ptr++;

}

ptr = arr;

printf("In Accessing array values Using pointer");

for(i=0; i<5; i++)

{

printf("%3d", *ptr);

MAY							2018
W	M	T	W	T	F	S	S
18		1	2	3	4	5	6
19	7	8	9	10	11	12	13
20	14	15	16	17	18	19	20
21	21	22	23	24	25	26	27
22	28	29	30	31	-	-	-

pt++;

}

printf("\n Values accessing directly:");

for(i=0; i<5; i++)

{

printf("%3d", arr[i]);

}

printf("\n");

}

21 Saturday
APRIL

111-254 Week 16

// C program for pointer to function //

#include <stdio.h>

int add(int a, int b);

void main()

{ int (*ptr) (int, int);

int r1;

printf("main begins \n");

ptr = &add;

r1 = (*ptr) (10, 20);

printf("R1 = %d \n", r1);

22 Sunday

112-253 Week 16

printf("main ends \n");

} int add(int a, int b)

{ printf("from F1 \n");
return a+b;

}

MAY 2018						
W	M	T	W	T	F	S
18	1	2	3	4	5	6
19	7	8	9	10	11	12
20	14	15	16	17	18	19
21	21	22	23	24	25	26
22	28	29	30	31	-	-

2018

9 /* String Manipulation */

#include <stdio.h>

#include <string.h>

void main()

{
char str1[15], str2[15], str3[10];

int n, c, len, i;

clrscr();

printf("\n Enter the string 1");

gets(str1);

puts(str1);

printf("\n Enter the string 2");

gets(str2);

puts(str2);

printf("\n Enter the string 3");

scanf("%s", str3);

printf("%s", str3);

printf("\n *****");

Case 2: $c = strcmp(str1, str2);$
if ($c == 0$)

printf("\n Both are Equal ");

else

printf("\n Both are different");

break;

Case 4:

printf("\n The resultant string is: %s",
strcat(str1, str2));

break;

Case 5: for($i=0; i < strlen(str1); i++$)

$str1[i] = toupper(str1[i]);$

printf("%s", str1);

break;

default: printf("\n Enter correct choice");

}
}

08

09

10

11

12

13

15

16

17

18

Program 10: Programs on structure and unions.

```
#include <stdio.h> // program of structure
```

```
struct student {
```

```
    char firstName[50];
```

```
    int roll;
```

```
    float marks;
```

```
} s[10];
```

```
int main()
```

```
{  
    int i;
```

```
    printf("Enter information of students: \n");
```

```
    for(i=0; i<5; i++)
```

```
    {  
        s[i].roll = i+1;
```

```
        printf("\n For roll number %d \n",  
               s[i].roll);
```



```

08 printf("Enter first name");
09 scanf("%s", s[i].firstName);
10
11 printf("Enter marks: ");
12 scanf("%f", &s[i].marks);
13 }
14 printf("Displaying Information\n\n");
15 for(i=0; i<5; i++)
16 {
17     printf("\n Roll Number: %d\n", i+1);
18     printf("first Name: ");
19     puts(s[i].firstName);
20     printf("Marks: %.1f", s[i].marks);
21     printf("\n");
22 } return 0;
23 }

```

MAY		2018						
W	M	T	W	T	F	S	S	
18		1	2	3	4	5	6	
19	7	8	9	10	11	12	13	
20	14	15	16	17	18	19	20	
21	21	22	23	24	25	26	27	
22	28	29	30	31	-	-	-	

// C program for Unions //

#include <stdio.h>

#include <string.h>

Union employee

{

int id;

char name[50];

}

e1;

int main()

{

e1.id = 101;

strcpy(e1.name, "NAWABS");

printf("employee 1 id is %d\n", e1.id);

printf("employee 1 name: %s\n", e1.name);

return 0;

}

Program 11: Finding the number of characters, words and lines of given text file!

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

```
void main()
```

```
{ char line[81], chr;
```

```
int i, c, end=0, characters=0, words=0,
lines=0;
```

```
printf("KEY IN THE TEXT");
```

```
printf("GIVE ONE SPACE AFTER EACH WORD
WHEN COMPLETED, PRESS RETURN");
```

```
while(end==0)
```

```
{ c=0;
```

```
while((chr=getchar()) != '\n')
```

```
line[c++] = chr;
```

```
line[c] = '\0';
```

```
if(line[0] == '\0')
```

```
break;
```

MAY							2018
W	M	T	W	T	F	S	S
18		1	2	3	4	5	6
19	7	8	9	10	11	12	13
20	14	15	16	17	18	19	20
21	21	22	23	24	25	26	27
22	28	29	30	31	-	-	-


```
else
```

```
{  
    words++;
```

```
for(i=0; line[i]!='\0'; i++)
```

```
if (line[i] == ' ' || line[i] == '\t')
```

```
    words++;
```

```
}
```

```
lines = lines + 1;
```

```
characters = characters + strlen(line);
```

```
}  
printf("\n");
```

```
printf("number of lines = %d", lines);
```

```
printf("number of words = %d", words);
```

```
printf("number of characters = %d", characters);
```

```
}
```

APRIL							2018
W	M	T	W	T	F	S	S
13	30						1
14	2	3	4	5	6	7	8
15	9	10	11	12	13	14	15
16	16	17	18	19	20	21	22
17	23	24	25	26	27	28	29

Program 12: File Handling Programs.

// C program to a-textfile using fprintf() //

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

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

```
void main(void)
```

```
{
```

```
int num;
```

```
FILE *fptr;
```

```
fptr = fopen("c:\\program.txt", "w");
```

```
if (fptr == NULL)
```

```
{ printf("Error!");
```

```
exit(1);
```

```
}
```

```
printf("Enter num");
```

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

```
fprintf(fptr, "%d", num);
```

```
fclose(fptr);
```

```
}
```

29 Sunday

MAY 2018						
W	M	T	W	T	F	S
18	1	2	3	4	5	6
19	7	8	9	10	11	12
20	14	15	16	17	18	19
21	21	22	23	24	25	26
22	28	29	30	31	-	-

08
09 // Read from a text file using fscanf() //

10 #include <stdio.h>

11 #include <stdlib.h>

void main(void)

12 {
int num;

13 FILE *fptr;

15 if ((fptr = fopen("C:\\program.txt", "r"))
== NULL)

16 {
printf("Error! Opening file");
exit(1);

17 }
18 fscanf(fptr, "%d", &num);

printf("Value of n = %d", num);

fclose(fptr);

}