

SUM OF THREE NUMBERS :-

Aim :- A C program to find sum of three values

Program :-

```
/* To find sum of three values */
#include <stdio.h>
#include <conio.h>
void main()
{
    int a,b,c, sum;
    clrscr();
    printf("enter the values of a,b,c : ");
    scanf ("%d %d %d", &a,&b,&c);
    sum = a+b+c;
    printf("a=%d\nb=%d\nc=%d \n sum=%d", a,b,c,sum);
    getch();
}
```

Input & Output :-

① enter the values of a,b,c : 15 20 10

a=15

b=20

c=10

sum=45

② enter the values of a,b,c : 5 8 2

a=5

b=8

c=2

sum=15

MAXIMUM OF TWO NUMBERS :-

Aim:- A C-program to find maximum of two numbers.

Program:-

```
/*maximum of two numbers */  
#include<stdio.h>  
#include <conio.h>  
void main()  
{  
    int a, b, max;  
    clrscr();  
    printf("max of a and b:");  
    scanf("%d %d", &a, &b);  
    max = (a>b)?a:b;  
    printf("max of two values max=%d", a, b, max);  
    getch();  
}
```

Input & Output :-

① max of a,b : 5 7

max of two values max= 7

② max of a,b : 81 19

max of two values max = 81

Result :-

Hence a C-program with given input are executed successfully and expected output is obtained.

10
10

28/6/18

EXCHANGE OF TWO NUMBERS :-

Aim:- A C-program to exchange of two numbers.

Program:-

```
/*exchange of two numbers */
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b,c;
    clrscr();
    printf("enter two values a and b:");
    scanf("%d %d", &a, &b);
    printf("before swapping:");
    printf("a=%d , b=%d", a,b);
    c=a
    a=b
    b=c
    printf("\n after swapping:");
    printf("a=%d , b=%d , c=%d", a,b, c=(a=b, b=c));
    getch();
}
```

Input & Output:-

① enter two values a and b : 5 8

before swapping : a=5 , b=8

after swapping : a=8 , b=5 , c=5

② enter two values a and b : 9 10

before swapping : a=9 , b=10

a=10 , b=9

DIFFERENT DATA TYPES ALONG WITH THEIR SIZES :-

Aim :- A C-program to print data values along with their sizes.

Program :-

```
/* to print data values along with sizes */
#include<stdio.h>
#include<conio.h>
void main()
{
    char a;
    int b;
    float c;
    double d;
    clrscr();
    printf("enter the values of char ,integer ,float and
           double : ");
    scanf("%c%d%f%lf", &a, &b, &c, &d);
    printf("\n a=%c, size = %d bytes", a, sizeof(a));
    printf("\n b=%d, size = %d bytes", b, sizeof(b));
    printf("\n c=%f, size = %d bytes", c, sizeof(c));
    printf("\n d=%lf, size = %d bytes", d, sizeof(d));
    getch();
}
```

PRIORITY AND ASSOCIATIVITY:-

Aim:- A C-program to print priority and associativity of operators.

Program:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a,b,c, priority, associativity;
    clrscr();
    printf("enter the values of a,b and c:");
    scanf("%d %d %d", &a, &b, &c);
    priority = a * b + c;
    associativity = a - b + c;
    printf("priority is %d in associativity is %d", priority,
           associativity);
    getch();
}
```

Input and Output:

- (1) enter the values of a,b and c: 4 3 2
priority is 14
associativity is 3
- (2) enter the values of a,b and c : 30 25 42

FACTORIAL OF A NUMBERS :-

Aim:- A C-program to find factorial of any number.

Program:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i, n, fact = 1;
    clrscr();
    printf("enter the value of n:");
    scanf("%d", &n);
    for(i=1; i<=n; i++)
        fact = fact * i;
    printf("fact = %d", fact);
    getch();
}
```

Input and Output:-

① enter the value of n: 3

fact = 6

② enter the value of n: 4

fact = 24.

Result:- Hence a C-program for given inputs are executed successfully and expected outputs are obtained.

Q. Q. Q.

PRIME OR NOT :-

Aim:- A C program to find the number is prime or not

Program:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i, n, count=0;
    clrscr();
    printf("enter the value of n:");
    scanf("%d", &n);
    for(i=1; i<=n; i++)
    {
        if(n%i==0)
            count++;
    }
    if(count==2)
        printf("%d is prime", n);
    else
        printf("%d is not prime", n);
    getch();
}
```

Exp:-11

12/7/18

SERIES OF PRIME NUMBERS:-

Aim:- A C- program to find series of prime numbers

Program:-

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int n, i, count=0, j;
    clrscr();
    printf("enter a value of n:");
    scanf("%d", &n);
    for(i=1; i<=n; i++)
    {
        count=0;
        for(j=1; j<=i; j++)
        {
            if(i%j==0)
                count++;
        }
        if(count==2)
            printf("\n%d", i);
    }
    getch();
}
```

Input and output:-

① enter a value of n: 5

2

3

4

5

Exp:-12

19/7/18

SUM OF INDIVIDUAL DIGITS:-

Aim:- A C program to find sum of individual digits of a given value.

Program:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n, r, sum=0, p,
    clrscr();
    printf("enter the value of n:");
    scanf("%d", &n);
    p=n;
    while(n>0)
    {
        r=n%10;
        sum=sum+r;
        n=n/10;
    }
    printf("sum of the digits in %d is %d", p, sum);
    getch();
}
```

REVERSING OF DIGITS :-

Aim:- The C-program to find the reversing the digits of a given number.

Program:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n, reversednumber=0, remainder;
    clrscr();
    printf("enter an integer:");
    scanf("%d", &n);
    remainder=n;
    while(n!=0)
    {
        remainder = n%10;
        reversednumber = reversednumber*10 + remainder;
        n/=10; // or n=n/10;
    }
    printf("reversednumber = %d", reversed number);
    getch();
}
```

PALINDROME OR NOT :-

Aim:- A C-program to check the input value is palindrome or not.

Program:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n, reversedinteger=0, remainder, originalinteger;
    clrscr();
    printf("enter an integer:");
    scanf("%d", &n);
    originalinteger=n;
    while(n!=0)
    {
        remainder=n%10;
        reversedinteger=reversedinteger*10 + remainder;
        n/=10;
    }
    if(originalinteger == reversedinteger)
        printf("the reverse of %d is %d , it is a palindrome",
               originalinteger, reversedinteger);
    else
        printf("the reverse of %d is not %d, it is not a palindrome",
               originalinteger, reversedinteger);
    getch();
}
```

ARMSTRONG OR NOT :-

Aim:- A C-program to find a number is Armstrong or not.

Program:-

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
    int number, sum=0, rem, cube, temp;
    clrscr();
    printf("enter a number:");
    scanf("%d", &number);
    temp = number;
    while(number != 0)
    {
        rem = number % 10;
        cube = pow(rem, 3);
        sum = sum + cube;
        number = number / 10;
    }
    if(sum == temp)
        printf("the given number %d is an armstrong number", temp);
    else
        printf("the given number %d is not an armstrong number", temp);
    getch();
}
```

Exp:-16

26/7/18

FIBBONACCI SERIES :-

Aim:- A C program to find the fibonacci series numbers upto a limit.

Program:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int f1=0, f2=1, f3, i=3, len;
    clrscr();
    printf("enter the length of fibonacci series:");
    scanf("%d", &len);
    printf("the fibonacci series is:");
    printf("\n%d\n%d", f1, f2);
    while(i<=len)
    {
        f3=f1+f2;
        printf("\n%d", f3);
        f1=f2;
        f2=f3;
        i++;
    }
    getch();
}
```

Sum of Elements of An Array.

Aim:- A C-program to find the sum of elements of an array.

Program:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[5], i, sum=0;
    clrscr();
    printf("enter 5 array elements :");
    for(i=1; i<=5; i++)
        scanf("%d", &a[i]);
    for(i=1; i<=5; i++)
    {
        sum = sum + a[i];
    }
    printf("\nthe array elements are:");
    for(i=1; i<=5; i++)
        printf("\n%d", a[i]);
    printf("\nsum of array elements=%d", sum);
    getch();
}
```

MAXIMUM AND MINIMUM AMONG A SET OF NUMBERS

Aim:- A C-program to find maximum and minimum among a set of numbers.

Program:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[5], i, max, min;
    clrscr();
    printf("enter the 5 array elements:");
    for (i=1; i<=5; i++)
        scanf("%d", &a[i]);
    printf("\n the array elements are:");
    for(i=1; i<=5; i++)
        printf("\n%d", a[i]);
    max=min=a[i];
    for(i=2; i<=5; i++)
    {
        if(a[i]>max)
            max=a[i];
        if(a[i]<min)
            min=a[i];
    }
    printf("\n max=%d\n min=%d", max, min);
    getch();
}
```

Sum of Positive and Negative Elements in a Matrix:-

Aim:- A C-program to find the sum of positive and negative elements of a matrix.

Program:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[3][3], i, j, pos=0, neg=0;
    clrscr();
    printf("enter 3*3 matrix elements:");
    for(i=0; i<=3; i++)
    {
        for(j=0; j<=3; j++)
            scanf("%d", &a[i][j]);
    }
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
        {
            if(a[i][j] >=0)
                pos = pos + a[i][j];
            else
                neg = neg + a[i][j];
        }
    }
    printf("sum of pos=%d \n sum of neg=%d", pos, neg);
    getch();
}
```

Sum of Matrices :-

Aim:- A C-program to find the sum of 2 matrices.

Programs:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[2][2], b[2][2], c[2][2], i, j;
    clrscr();
    printf("enter 2*2 matrix elements:");
    for(i=1; i<=2; i++)
    {
        for(j=1; j<=2; j++)
            scanf("%d", &a[i][j]);
    }
    printf("\n enter 2*2 matrix elements:");
    for(i=1; i<=2; i++)
    {
        for(j=1; j<=2; j++)
            scanf("%d", &b[i][j]);
    }
    printf("the 2*2 matrix elements of a:");
    for(i=1; i<=2; i++)
    {
        printf("\n");
        for(j=1; j<=2; j++)
            printf("%d\t", a[i][j]);
    }
}
```

```

printf("\n the 2*2 matrix elements of b:");
for(i=1; i<=2; i++)
{
    printf("\n");
    for(j=1; j<=2; j++)
        printf("%d\t", b[i][j]);
}
printf("\n the addition of 2*2 matrix:");
c[2][2]=0;
for(i=1; i<=2; i++)
{
    printf("\n");
    for(j=1; j<=2; j++)
    {
        c[i][j] = a[i][j] + b[i][j];
        printf("%d\t", c[i][j]);
    }
}
getch();
}

```

Input and output:-

① enter 2*2 matrix element : 1 1 1
 enter 2*2 matrix element : 1 1 1
 the 2*2 matrix elements of a:

1 1
 1 1

..... sum of elements of b.

PRODUCT OF MATRICES

Aim:- A C-program to find the product of two matrices.

Program:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[2][2], b[2][2], c[2][2], i, j, k;
    clrscr();
    printf("enter 2*2 matrix elements : ");
    for(i=1; i<=2; i++)
    {
        for(j=1; j<=2; j++)
            scanf("%d", &a[i][j]);
    }
    printf("\n enter 2*2 matrix elements : ");
    for(i=1; i<=2; i++)
    {
        for(j=1; j<=2; j++)
            scanf("%d", &b[i][j]);
    }
    printf("the 2*2 matrix elements of a : ");
    for(i=1; i<=2; i++)
    {
        printf("\n");
        for(j=1; j<=2; j++)
            printf("%dit", a[i][j]);
    }
}
```

```
{  
printf("\n the 2*2 matrix elements of b:");  
for(i=1; i<=2; i++)  
{  
printf("\n");  
for(j=1; j<=2; j++)  
printf("%d", b[i][j]);  
}  
for(i=1; i<=2; i++)  
{  
printf("\n");  
for(j=1; j<=2; j++)  
{  
c[i][j]=0;  
for(k=1; k<=2; k++)  
{  
c[i][j]=c[i][j]+a[i][k]*b[k][j];  
}  
}  
printf("\n the product of two matrices!);  
for(i=1; i<=2; i++)  
{  
printf("\n");  
for(j=1; j<=2; j++)  
printf("%3d", c[i][j]);  
}  
getch();  
}
```

TRANSPOSE OF MATRIX :-

Aim:- A C-program to find the transpose of a matrix.

Program:-

```
#include<stdio.h>
#include<conio.h>
Void main()
{
    int a[3][3], i, j;
    clrscr();
    printf("enter 3*3 matrix elements : ");
    for(i=1; i<=3; i++)
    {
        for(j=1; j<=3; j++)
            scanf("%d", &a[i][j]);
    }
    printf("\nthe array elements are : ");
    for(i=1; i<=3; i++)
    {
        printf("\n");
        for(j=1; j<=3; j++)
            printf("%.3d", a[i][j]);
    }
    printf("\nthe transpose of given matrix is : ");
    for(i=1; i<=3; i++)
    {
        printf("\n");
        for(j=1; j<=3; j++)
            printf("%.3d", a[j][i]);
    }
    getch();
}
```

Sum of Even and Odd Elements of a Matrix :-

Aim:- A C-program to find sum of even and odd elements of a matrix.

Program:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[3][3], i, j, even=0, odd=0,
        clrscr();
    printf("enter 3*3 matrix elements :");
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
            scanf("%d", &a[i][j]);
    }
    printf("array elements are \n");
    for(i=0; i<3; i++)
    {
        printf("\n");
        for(j=0; j<3; j++)
            printf("%.3d", a[i][j]);
    }
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
        {
            if(a[i][j] % 2 == 0)
                even = even + a[i][j];
            else
                odd = odd + a[i][j];
        }
    }
}
```

MUTUAL

```
    }  
    printf("sum of even=%d \n sum of odd=%d", even, odd);  
    getch();  
}
```

Input and Output:-

① enter 3*3 matrix elements : 1 2 3 4 5 6 7 8 9
array elements are

1	2	3
4	5	6
7	8	9

Sum of Diagonal Elements of a Matrix :-

Aim:- A C-program to find a sum of diagonal elements of a matrix.

Program:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[3][3], i, j, m, n, sum=0;
    clrscr();
    printf("enter m and n values : \n");
    scanf("%d %d", &m, &n);
    printf("enter %d elements : ", m * n);
    for(i=0; i<m; i++)
    {
        for(j=0; j<n; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }
    printf("array elements are \n");
    for(i=0; i<m; i++)
    {
        printf("\n");
        for(j=0; j<n; j++)
        {
            printf("%3d", a[i][j]);
        }
    }
    for(i=0; i<m; i++)
    {
        for(j=0; j<n; j++)
        {
            if(i==j)
                sum = sum + a[i][j];
        }
    }
    printf("\n sum of diagonal matrix = %d", sum);
}
```

MATRIX :-

```
getch();
```

```
}
```

Input and Output:-

① Enter m and n values :-

3 3

Enter m x n matrix :-

SORTING OF ELEMENTS:-

Aim:- A C-program to sort the elements in the ascending order.

Program:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[5], i, j, temp;
    clrscr();
    printf("enter 5 elements : ");
    for(i=1; i<=5; i++)
        scanf("%d", &a[i]);
    printf("\nthe array elements are : ");
    for(i=1; i<=5; i++)
        printf("%d\t", a[i]);
    for(i=1; i<=5; i++)
    {
        for(j=i+1; j<=5; j++)
        {
            if(a[i] > a[j])
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
}
```

```
printf("In the sorted array elements are:");
for(i=1; i<=5 ; i++)
printf("\n%d",a[i]);
getch();
}
```

Input and Output:-

① enter 5 elements : 2 5 1 6 4

the array elements are :

2

5

1

6

4

the sorted array elements are :

1

2

4

SORTING OF STRINGS :-

Aim:- A C-program of sorting of strings in an alphabetical order.

Program :-

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char a[10][10], temp[10];
    int i, j;
    clrscr();
    printf("enter to 10 characters:");
    for(i=0; i<=10; i++)
        scanf("%s", &a[i]);
    for(i=0; i<9; i++)
    {
        for(j=i+1; j<10; j++)
        {
            if(strcmp(a[i], a[j]) > 0)
            {
                strcpy(temp, a[i]);
                strcpy(a[i], a[j]);
                strcpy(a[j], temp);
            }
        }
    }
    printf("\nthe sorted characters are:");
    for(i=0; i<=10; i++)
        printf("%s\t", a[i]);
    getch();
}
```

FACTORIAL OF A NUMBER BY RECURSION:-

Aim:- A C-program to find the factorial of a number by recursion.

Program:-

```
#include<stdio.h>
#include<conio.h>
factorial (int);
void main()
{
    int n,f;
    clrscr();
    printf("enter n value:");
    scanf ("%d", &n);
    f=factorial(n);
    printf("\n factorial of %d is %d", n,f);
    getch();
}
factorial (int n)
{
    if(n==1)
        return 1;
    else
        return (n*factorial (n-1));
}
```

ROOTS OF QUADRATIC EQUATION

Aim:- A C program to find the roots of a quadratic equation.

Program:-

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
    int a,b,c;
    float d, r1,r2;
    clrscr();
    printf("enter a,b and c values:");
    scanf("%d%d%d", &a, &b, &c);
    if(a==0)
    {
        printf("equation is linear");
        d=-c/b;
        printf("the root is %.f", d);
    }
    else
    {
        d=(b*b)-4*a*c;
        if(d==0)
        {
            printf("roots are equal");
            d=-b/2*a;
            printf("root 1 = %.f, root 2 = %.f", r1, r2);
        }
        else
        {
            if(d>0)
            {
                printf("roots are different");
                r1=(-b+sqrt((b*b)-4*a*c))/(2*a);
                r2=(-b-sqrt((b*b)-4*a*c))/(2*a);
                printf("the roots of %.f and %.f", r1, r2);
            }
        }
    }
}
```

```
{  
    printf("The roots are imaginary");  
    getch();  
}
```

Input and Output :-

- ① enter a,b and c values : 3 4 5
the roots are imaginary.
- ② enter a,b and c values : 1 2 3
the roots are imaginary.

SWAPPING OF TWO VALUES By CALL By REFERENCE

Aim: A C program to swap two values by using call by reference of function with argument and no return values.

Program:-

```
#include<stdio.h>
#include<conio.h>
void swap(int*, int*);
void main()
{
    int a, b;
    clrscr();
    printf("enter a, b values before swapping:");
    scanf("%d%d", &a, &b);
    swap(a, b);
    getch();
}

void swap(int*a, int*b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
    printf("after swapping a=%d, b=%d", *a, *b);
}
```

GCD USING RECURSION:-

Aim:- A C-program to calculate the GCD of two integer values.

Program:-

```
#include<stdio.h>
#include<conio.h>
GCD(int x, int y)
{
    if(x==y)
        return x;
    else if(x>y)
        return (GCD((x-y), y));
    else
        return (GCD(x, (y-x)));
}
void main()
{
    int a,b,gcd;
    clrscr();
    printf("enter the values of a and b:");
    scanf("%d %d", &a, &b);
    gcd=GCD(a,b);
    printf("The gcd of %d,%d is %d", a, b, gcd);
    getch();
}
```

SUM OF SERIES :-

Aim:- A C program to find sum of series of for following equation.

$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

Program:-

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

fact (int x)
{
    if (x==0 || x==1)
        return 1;
    else
        return (x* fact(x-1));
}

void main()
{
    int x,n,i,sum=1;
    clrscr();
    printf("enter the values of x and n:");
    scanf("%d %d", &x, &n);
    for(i=1; i<=n; i++)
        sum = sum + (pow(x,i) / fact(i));
    printf("The sum of series is %d", sum);
}
```

```
getch();  
}  
}
```

Input and Output:-

- ① enter the values of x
the sum of series
- ② enter the values of x
the sum of series

To COUNT NO.OF VOWELS , CONSONANTS, WHITE SPACES, DIGITS AND SPECIAL CHARACTERS :-

Aim :- A C-program to find no.of vowels, consonants, white spaces, digits and speci characters in a given string.

Program:-

```
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
void main()
{
    char s[10];
    int i, v=0, c=0, d=0, sp=0, sc= 0;
    clrscr();
    puts("enter a string :");
    gets(s);
    for(i=0; s[i]!='\0'; i++)
    {
        s[i]=tolower(s[i]);
        if(isalpha(s[i])!=0)
        {
            if(s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' || s[i]=='u')
                v++;
            else
                c++;
        }
        else
            d++;
        if(s[i]==' ')
            sp++;
        if(s[i]==',')
            sc++;
    }
    printf("No. of vowels = %d\n", v);
    printf("No. of consonants = %d\n", c);
    printf("No. of digits = %d\n", d);
    printf("No. of white spaces = %d\n", sp);
    printf("No. of special characters = %d", sc);
}
```

```
{  
else if(isdigit(s[i]) != 0)  
    d++;  
else if(isSpace(s[i]) != 0)  
    sp++;  
else  
    sc++;  
  
printf("no. of vowels = %d", v);  
printf("no. of consonants = %d", c);  
printf("no. of digits = %d", d);  
printf("no. of spaces = %d", sp);  
printf("no. of special characters = %d", sc);  
getch();  
}
```

Input and Output :-

① enter a string :

naveen KUMAR 56@

no. of vowels = 5

no. of consonants = 6

no. of digits = 2

PASCAL TRIANGLE :-

Aim:- A C-program to generate pascal triangle for the given no. of rows.

Program :-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int row, b=1, x, y, z;
    clrscr();
    printf("enter the no. of rows : ");
    scanf("%d", &row);
    printf("\nthe pascal triangle is :\n");
    for(y=0; y<row; y++)
    {
        for(x=40-3*y; x>0; x--)
            printf(" ");
        for(z=0; z<=y; z++)
        {
            if(y==0 || z==0)
                b=1;
            else
                b=(b*(y-z+1))/z;
            printf("%6d", b);
        }
        printf("\n");
    }
}
```

```
    getch();  
}
```

Inputs and Outputs:-

① Enter the no. of rows of the pascal triangle

BASE CONVERSION

Aim:- A C - program to convert values into different bases.

Program:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n;
    clrscr();
    printf("enter the value of n:");
    scanf("%d", &n);
    printf("decimal=%d", n);
    printf("\noctal =%o", n);
    printf("\nhexagonal=%x", n);
    printf("\nvalue=%i", n);
    getch();
}
```

Exp: 35

G10118

MULTIPLICATION TABLES

FROM 11 to 20

Aim:- A C-program to print multiplication tables from 11 to 20.

Program:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,j;
    char c;
    clrscr();
    printf("the multiplication tables from 11 to 20:");
    for(i=11; i<=20; i++)
    {
        for(j=1; j<=10; j++)
            printf("\n %d * %d = %d", i, j, i*j);
        c=getchar();
    }
    getch();
}
```

RANDOM NUMBER GENERATION :-

Aim:- A C-program to generate random numbers.

Program:-

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

void main()
{
    int i;
    clrscr();
    printf("the random numbers are :");
    for(i=1; i<=5; i++)
        printf("\nIn the %d random numbers is %d", i, rand());
    getch();
}
```



LENGTH OF STRINGS

Aim:- A C-program to find the no. of characters in a string.

Program:-

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char s[20];
    int i, count=0;
    clrscr();
    puts("enter a string :");
    gets(s);
    for(i=0; s[i]!='\0'; i++)
        count++;
    printf("In the %s length is %.d", s, count);
    getch();
}
```



COMPARING TWO STRINGS

Aim: A C-program for comparing two strings

Program:

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

void main()
{
    char s1[20], s2[20];
    int i, sum1=0, sum2=0;
    clrscr();
    puts("enter string 1:");
    gets(s1);
    puts("enter string 2:");
    gets(s2);
    for(i=0; s1[i]!='\0'; i++)
        sum1 = sum1 + s1[i];
    for(i=0; s2[i]!='\0'; i++)
        sum2 = sum2 + s2[i];
    if(sum1 > sum2)
        puts("string 1 is greater than string 2");
    else if(sum1 < sum2)
        puts("string 1 is less than string 2");
    else
```

```
printf("strings      are      identical");  
getch();  
}
```

Input and Output :-

① enter string 1:

naveen

enter string 2:

naveen

strings are identical.

② enter string 1:

computer

enter string 2:

Exp:-39

11/11/18

Reversing A String:-

Aim:- A C-program for reversing the set of characters of a string

Program:-

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char s[20], t[20];
    int i, j;
    clrscr();
    puts("enter string :");
    gets(s);
    for(i=0; s[i]!='\0'; i++);
    for(j=0; i>0; j++)
        t[j] = s[--i];
    t[j] = '\0';
    puts(s);
    puts("the reverse string is :");
    puts(t);
    getch();
}
```

```
void main()
{
    printf("welcome navi");
    getch();
}

{
    r=0;
    int n, rev remainder;
    printf("Enter an integer:");
    scanf("%d", &n);
    while (n!=0)
    {
        remainder = n%10;
        rev = rev + 10 + remainder;
        n /= 10;
    }
    printf("Reverse = %d", rev);
    return 0;
}
```

$$\begin{aligned} r &= r + 10 \\ r &= r + n \% 10 \\ n &= n / 10 \end{aligned}$$

Exp: 40

11/11/18

DELETING CHARACTERS IN A STRING :-

Aim:- A C-program to delete the characters in a string.

Program:-

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

void main()
{
    char s[20];
    int position;
    puts("enter the string :");
    gets(s);
    printf("enter the position from where character\n"
           "have to be deleted :");
    scanf("%d", &position);
    s[position] = '\0';
    puts(s);
    getch();
}
```

CONCATENATION OF TWO STRINGS :-

Aim:- A C-program to add the second string into first string.

Program :-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char s1[20], s2[20];
    int i, j;
    clrscr();
    puts("enter string 1:");
    gets(s1);
    puts("enter string 2:");
    gets(s2);
    puts(s1);
    puts(s2);
    for(i=0; s1[i]!='\0'; i++);
    {
        for(j=0; s2[j]!='\0'; j++, i++)
            s1[i]=s2[j];
    }
    s1[i]='\0';
    printf("\n the concatenation string is %.s", s1);
```

```
getch();
```

```
}
```

Input and Output :-

① Enter string 1 :

Computer

Enter string 2 :

Programming

Computer

Programming

COMPLEX NUMBER OPERATIONS

Aim → A C program to perform add, sub, and multiplication on complex numbers.

Program:

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

void add(struct complex, struct complex);
void sub(struct complex, struct complex);
void mul(struct complex, struct complex);

struct complex
{
    float r;
    float i;
} c1, c2, c3;

void main()
{
    clrscr();

    printf("enter first complex number real and\nimaginary part:");
    scanf("%f %f", &c1.r, &c2.i);

    printf("enter second complex number real and\nimaginary part:");
    scanf("%f %f", &c2.r, &c2.i);

    add(c1, c2);
    sub(c1, c2);
    mul(c1, c2);
}
```

```

getch();
3 void addCstruct complex c1, struct complex c2)
{
c3.r = c1.r + c2.r;
c3.i = c1.i + c2.i;
printf("In sum=(%.2f)+i(%.2f)", c3.r, c3.i);
3 void subCstruct complex c1, struct complex c2)
{
c3.r = c1.r - c2.r;
c3.i = c1.i - c2.i;
printf("In sub=(%.2f)+i(%.2f)", c3.r, c3.i);
3 void mul(struct complex c1, struct complex c2)
{
c3.r = (c1.r * c2.r) - (c1.i * c2.i);
c3.i = (c1.r * c2.i) + (c1.i * c2.r);
printf("In mul=(%.2f)+i(%.2f)", c3.r, c3.i);
}

```

Input and Output:-

① enter first complex number real and imaginary part : 2 3
 enter second complex number real and imaginary part : 3 2
 $\text{sum} = (5.00) + i(5.00)$
 $\text{sub} = (-1.00) + i(1.00)$
 $\text{mul} = (0.00) + i(13.00)$

STUDENT GRADE USING STRUCTURE

Aim:- A C-program to calculate the grade of a student by using structures.

Program:-

```
#include<stdio.h>
#include<conio.h>
struct student
{
    int rno;
    int marks[6];
} s;
void main()
{
    int i, total = 0, avg;
    clrscr();
    printf("enter the student rno and 6 subjects marks:");
    scanf("%d", &s.rno);
    for (i = 1; i <= 6; i++)
    {
        scanf("%d", &s.marks[i]);
        total = total + s.marks[i];
    }
    avg = total / 6;
    printf("total = %.d, avg = %.d", total, avg);
    if (s.marks[1] < 35 || s.marks[2] < 35 || s.marks[3] < 35 || s.marks[4]
        < 35 || s.marks[5] < 35 || s.marks[6] < 35)
```

```
printf("In FAIL");
else if(avg >= 75)
printf("In DISTINCTION");
else if(avg >= 60)
printf("In FIRST CLASS");
else if(avg >= 50)
printf("In SECOND CLASS");
else
printf("In THIRD CLASS");
getch();
}
```

Input and Output :-

① Enter the student rno 6
1

MERGE FILES :-

Aim:- A C-program to merge two files.

Program:-

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

void main()
{
    FILE *fp1, *fp2, *fp3;
    char c;
    clrscr();
    fp1 = fopen("file1.txt", "w+");
    printf("press * to stop:");
    while ((c = getchar()) != '*')
        putc(c, fp1);
    printf("\npress * to stop:");
    fp2 = fopen("file2.txt", "w+");
    while ((c = getchar()) != '*')
        putc(c, fp2);
    rewind(fp1);
    rewind(fp2);
    fp3 = fopen("file3.txt", "w+");
    while ((c = fgetc(fp1)) != EOF)
        putc(c, fp3);
    while ((c = fgetc(fp2)) != EOF)
```

```
putc(c, fp3);
rewind(fp3);
printf("the contents of merged file are :\n");
while((c = fgetc(fp3)) != EOF)
    putchar(c);
getchar();
}

Inputs and Outputs:
    . . . . . *  
ctrl : naveen *
```