```
//1 C program to perform all arithmetic operations
#include<stdio.h>
int main()
{
 int a,b;
 printf("enter two numbers for operations");
 scanf("%d%d",&a,&b);
 int sum, sub,mult, mod;
 float div;
 sum=a+b;
 sub=a-b;
 mult=a*b;
 mod=a%b;
 div=a/b;
 printf("sum =%d\n",sum);
 printf("sub =%d\n",sub);
 printf("mult=%d\n",mult);
 printf("div =%f\n",div);
 printf("mod =%d\n",mod);
return 0;
}
```

OUTPUT

```
enter two numbers for operations20
10
sum=30
sub=10
mult=200
div=2.000000
mod=0
```

```
/**
 * C program to find area of a triangle if base and height are given
 */

#include <stdio.h>
int main()
{
    float base, height, area;
    printf("Enter base of the triangle: ");
    scanf("%f", &base);
    printf("Enter height of the triangle: ");
    scanf("%f", &height);

    area = (base * height) / 2;
    printf("Area of the triangle = %f sq. units", area);

    return 0;
}
```

OUTPUT

```
Enter base of the triangle:4
Enter height of the triangle:5
Area of the triangle = 10.000000 sq. units
```

```c
/**
 * C program to find all angles of a triangle if two angles are given
 */

#include <stdio.h>

int main()
{
    int a, b, c;

    /* Input two angles of the triangle */
    printf("Enter two angles of triangle: ");
    scanf("%d%d", &a, &b);

    /* Compute third angle */
    c = 180 - (a + b);

    /* Print value of the third angle */
    printf("Third angle of the triangle = %d", c);

    return 0;
}
```

OUTPUT

```
Enter two angles of triangle: 60 30
Third angle of the triangle = 90
```

```
//4 C program to convert days in to year,weeks and days
#include <stdio.h>
int main()
{
   int days, years, weeks;

   days = 1329;

   // Converts days to years, weeks and days
   years = days/365;
   weeks = (days % 365)/7;
   days = days- ((years*365) + (weeks*7));

   printf("Years: %d\n", years);
   printf("Weeks: %d\n", weeks);
   printf("Days: %d \n", days);

   return 0;
}
```

Output:

```
Years: 3
Weeks: 33
Days: 3
```

```
/**6
 * C program to calculate total, average and percentage of five subjects
 */

#include <stdio.h>

int main()
{
    float eng, phy, chem, math, comp;
    float total, average, percentage;

    /* Input marks of all five subjects */
    printf("Enter marks of five subjects: \n");
    scanf("%f%f%f%f%f", &eng, &phy, &chem, &math, &comp);

    /* Calculate total, average and percentage */
    total = eng + phy + chem + math + comp;
    average = total / 5.0;
    percentage = (total / 500.0) * 100;

    /* Print all results */
    printf("Total marks = %f\n", total);
    printf("Average marks = %f\n", average);
    printf("Percentage = %f", percentage);
    return 0;
}
```

```
/**7(A)MSB
 * C program to check Most Significant Bit (MSB) of a number using bitwise operator
 */

#include <stdio.h>
#define BITS sizeof(int) * 8 // Total bits required to represent integer

int main()
{
    int num, msb;

    /* Input number from user */
    printf("Enter any number: ");
    scanf("%d", &num);

    /* Move first bit of 1 to highest order */
    msb = 1 << (BITS - 1);

    /* Perform bitwise AND with msb and num */
    if(num & msb)
        printf("MSB of %d is set (1).", num);
    else
        printf("MSB of %d is unset (0).", num);

    return 0;
}
//7(B)LSB
/**
 * C program to check Least Significant Bit (LSB) of a number using bitwise operator
 */

#include <stdio.h>

int main()
{
    int num;

    /* Input number from user */
    printf("Enter any number: ");
    scanf("%d", &num);

    /* If (num & 1) evaluates to 1 */
    if(num & 1)
        printf("LSB of %d is set (1).", num);
    else
        printf("LSB of %d is unset (0).", num);

    return 0;
}
```

```
//8(A)swapping without using third variable
#include<stdio.h>
 int main()
{
int a=10, b=20;
printf("Before swap a=%d b=%d",a,b);
a=a+b;//a=30 (10+20)
b=a-b;//b=10 (30-20)
a=a-b;//a=20 (30-10)
printf("\nAfter swap a=%d b=%d",a,b);
return 0;
}
```

```
//8(B)swapping without using third variable
#include<stdio.h>
 int main()
{
int a=10, b=20;
printf("Before swap a=%d b=%d",a,b);
c=a;//a=10 store in c
a=b;//a=20 as a strored from b
b=c;//b=10 as stored from c
printf("\nAfter swap a=%d b=%d",a,b);
return 0;
}
```

```
//Maximum number using ternary(conditional)
# include <stdio.h>

int main()
{
    int a, b, c, big ;

    printf("Enter three numbers : ") ;

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

    big = a > b ? (a > c ? a : c) : (b > c ? b : c) ;

    printf("\nThe biggest number is : %d", big) ;
retuirn 0;
}
```

```
//10 Checking alphabates,digits and special character by conditional
#include <stdio.h>
int main()
{
    char ch;

    /* Input character from user */
    printf("Enter any character: ");
    scanf("%c", &ch);


    /* Alphabet check */
    if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))
    {
        printf("'%c' is alphabet.", ch);
    }
    else if(ch >= '0' && ch <= '9')
    {
        printf("'%c' is digit.", ch);
    }
    else
    {
        printf("'%c' is special character.", ch);
    }

    return 0;
}
```

```c
/**
 * 11 C program to calculate total electricity bill
 */

#include <stdio.h>

int main()
{
    int unit;
    float amt, total_amt, sur_charge;

    /* Input unit consumed from user */
    printf("Enter total units consumed: ");
    scanf("%d", &unit);


    /* Calculate electricity bill according to given conditions */
    if(unit <= 50)
    {
        amt = unit * 0.50;
    }
    else if(unit <= 150)
    {
        amt = 25 + ((unit-50) * 0.75);
    }
    else if(unit <= 250)
    {
        amt = 100 + ((unit-150) * 1.20);
    }
    else
    {
        amt = 220 + ((unit-250) * 1.50);
    }

    /*
     * Calculate total electricity bill
     * after adding surcharge
     */
    sur_charge = amt * 0.20;
    total_amt  = amt + sur_charge;

    printf("Electricity Bill = Rs. %f", total_amt);

    return 0;
}
```

```
//12(A) Simple calculator using switch case
#include <stdio.h>

int main()
{
    int num1,num2;
    float result;
    char ch;    //to store operator choice

    printf("Enter first number: ");
    scanf("%d",&num1);
    printf("Enter second number: ");
    scanf("%d",&num2);

    printf("Choose operation to perform (+,-,*,/,%): ");
    scanf(" %c",&ch);

    result=0;
    switch(ch)
    {
        case '+':
            result=num1+num2;
            break;

        case '-':
            result=num1-num2;
            break;

        case '*':
            result=num1*num2;
            break;

        case '/':
            result=(float)num1/(float)num2;
            break;

        case '%':
            result=num1%num2;
            break;
        default:
            printf("Invalid operation.\n");
    }

    printf("Result: %d %c %d = %f\n",num1,ch,num2,result);
    return 0;
}
```

Output and method

```
    First run:
    Enter first number: 10
    Enter second number: 20
    Choose operation to perform (+,-,*,/,%): +
    Result: 10 + 20 = 30.000000

    Second run:
```

```
    Enter first number: 10
    Enter second number: 3
    Choose operation to perform (+,-,*,/,%): /
    Result: 10 / 3 = 3.333333

    Third  time run:
    Enter first number: 10
    Enter second number: 3
    Choose operation to perform (+,-,*,/,%): >
    Invalid operation.
    Result: 10 > 3 = 0.000000
```

```
//12(B)Days of week
/**
 * C program to print day of week using switch case
 */

#include <stdio.h>

int main()
{
    int week;

    /* Input week number from user */
    printf("Enter week number(1-7): ");
    scanf("%d", &week);

    switch(week)
    {
        case 1:
            printf("Monday");
            break;
        case 2:
            printf("Tuesday");
            break;
        case 3:
            printf("Wednesday");
            break;
        case 4:
            printf("Thursday");
            break;
        case 5:
            printf("Friday");
            break;
        case 6:
            printf("Saturday");
            break;
        case 7:
            printf("Sunday");
            break;
        default:
            printf("Invalid input! Please enter week number between 1-7.");
    }

    return 0;
}
```

OUTPUT:

```
Enter week number(1-7): 1
Monday
```

```
//13 To check vowel and consonent
/**
 * C program to check vowel or consonant using switch case
 */

#include <stdio.h>

int main()
{
    char ch;

    /* Input an alphabet from user */
    printf("Enter any alphabet: ");
    scanf("%c", &ch);

    /* Switch value of ch */
    switch(ch)
    {
        case 'a':
            printf("Vowel");
            break;
        case 'e':
            printf("Vowel");
            break;
        case 'i':
            printf("Vowel");
            break;
        case 'o':
            printf("Vowel");
            break;
        case 'u':
            printf("Vowel");
            break;
        case 'A':
            printf("Vowel");
            break;
        case 'E':
            printf("Vowel");
            break;
        case 'I':
            printf("Vowel");
            break;
        case 'O':
            printf("Vowel");
            break;
        case 'U':
            printf("Vowel");
            break;
        default:
            printf("Consonant");
    }

    return 0;
}
```

```
//14 To check
/**
 * C program to check positive negative or zero using switch case
 */
#include <stdio.h>

int main()
{
    int num;

    printf("Enter any number: ");
    scanf("%d", &num);

    switch (num > 0)
    {
        // Num is positive
        case 1:
            printf("%d is positive.", num);
        break;

        // Num is either negative or zero
        case 0:
            switch (num < 0)
            {
                case 1:
                    printf("%d is negative.", num);
                    break;
                case 0:
                    printf("%d is zero.", num);
                    break;
            }
        break;
    }

    return 0;
}
```

```
// 15To check trangle property like Equilatral,isosceles,scalene
#include<stdio.h>
int main(){
  int side1, side2, side3;
  printf("Enter sides of triangle:");
  scanf("%d%d%d",&side1,&side2,&side3);
  if(side1 == side2 && side2 == side3)
    printf("The Given Triangle is equilateral");
  else if(side1 == side2 || side2 == side3 || side3 == side1)
    printf("The given Triangle is isosceles");
  else
    printf("The given Triangle is scalene");
  return 0;
}
```

```
/**
 *16 C program to find sum of natural numbers between 1 to n
 */

#include <stdio.h>

int main()
{
    int i, n, sum=0;

    /* Input upper limit from user */
    printf("Enter upper limit: ");
    scanf("%d", &n);

    /* Find sum of all numbers */
    for(i=1; i<=n; i++)
    {
        sum += i;
    }

    printf("Sum of first %d natural numbers = %d", n, sum);

    return 0;
}
```

```
/**17
 * C program to print sum of all even numbers between 1 to n
 */

#include <stdio.h>

int main()
{
    int i, n, sum=0;

    /* Input upper limit from user */
    printf("Enter upper limit: ");
    scanf("%d", &n);

    for(i=2; i<=n; i+=2)
    {
        /* Add current even number to sum */
        sum += i;
    }

    printf("Sum of all even number between 1 to %d = %d", n, sum);

    return 0;
}
```

```
//18 Multiplication of table
#include <stdio.h>
void main()
{
   int j,n;
   printf("Input the number (Table to be calculated) : ");
   scanf("%d",&n);
   printf("\n");
   for(j=1;j<=10;j++)
   {
     printf("%d X %d = %d \n",n,j,n*j);
   }
}
```

Output is like that:

```
Input the number (Table to be calculated) : 15

15 X 1 = 15
15 X 2 = 30
15 X 3 = 45
15 X 4 = 60
15 X 5 = 75
15 X 6 = 90
15 X 7 = 105
15 X 8 = 120
15 X 9 = 135
15 X 10 = 150
```

```
//19 Factorial of number
#include <stdio.h>
int main() {
    int n, i;
    unsigned long long fact = 1;
    printf("Enter an integer: ");
    scanf("%d", &n);

    // shows error if the user enters a negative integer
    if (n < 0)
        printf("Error! Factorial of a negative number doesn't exist.");
    else {
        for (i = 1; i <= n; ++i) {
            fact *= i;
        }
        printf("Factorial of %d = %llu", n, fact);
    }

    return 0;
}
```

Output like this:

```
Enter an integer: 10
Factorial of 10 = 10*9*8*7*6*5*4*3*2*1=3628800
```

```
//Palindrom checking...
#include <stdio.h>
int main() {
  int n, reversed = 0, remainder, original;
    printf("Enter an integer: ");
    scanf("%d", &n);
    original = n;

    // reversed integer is stored in reversed variable
    while (n != 0) {
        remainder = n % 10;
        reversed = reversed * 10 + remainder;
        n /= 10;
    }

    // palindrome if orignal and reversed are equal
    if (original == reversed)
        printf("%d is a palindrome.", original);
    else
        printf("%d is not a palindrome.", original);

    return 0;
}
```

```
/**21
 * C program to count frequency of digits in a given number
 */

#include <stdio.h>
#define BASE 10 /* Constant */

int main()
{
    long long num, n;
    int i, lastDigit;
    int freq[BASE];

    /* Input number from user */
    printf("Enter any number: ");
    scanf("%lld", &num);

    /* Initialize frequency array with 0 */
    for(i=0; i<BASE; i++)
    {
        freq[i] = 0;
    }

    /* Copy the value of 'num' to 'n' */
    n = num;

    /* Run till 'n' is not equal to zero */
    while(n != 0)
    {
        /* Get last digit */
        lastDigit = n % 10;

        /* Remove last digit */
        n /= 10;

        /* Increment frequency array */
        freq[lastDigit]++;
    }

    /* Print frequency of each digit */
    printf("Frequency of each digit in %lld is: \n", num);
    for(i=0; i<BASE; i++)
    {
        printf("Frequency of %d = %d\n", i, freq[i]);
    }

    return 0;
}
```

```
//LCM AND HCF
#include <stdio.h>
int main() {
  int a, b, x, y, t, gcd, lcm;

  printf("Enter two integers\n");
  scanf("%d%d", &x, &y);

  a = x;
  b = y;

  while (b != 0) {
    t = b;
    b = a % b;
    a = t;
  }

  gcd = a;
  lcm = (x*y)/gcd;

  printf("Greatest common divisor of %d and %d = %d\n", x, y, gcd);
  printf("Least common multiple of %d and %d = %d\n", x, y, lcm);

  return 0;
}
```

```
#include<stdio.h>

int main(){

    int num,i,count,n;
    printf("Enter max range: ");
    scanf("%d",&n);

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

         count = 0;

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

         if(count==0 && num!= 1)
             printf("%d ",num);
    }

   return 0;
}
```

OUTPUT:

```
Enter max range: 50
2 3 5 7 11 13
```

```
//24 Strong number
#include <stdio.h>
int main()
{
    int n;
    int sum=0;
    printf("Enter a number");
    scanf("%d",&n);
    int k=n;
    int r;
    while(k!=0)
    {
        r=k%10;
        int f=fact(r);
        k=k/10;
        sum=sum+f;
    }
    if(sum==n)
    {
        printf("\nNumber is a strong");
    }
    else
    {
        printf("\nNumber is not a strong");
    }
    return 0;
}
int fact(int r)
{
    int mul=1;
    for(int i=1;i<=r;i++)
    {
        mul=mul*i;
    }
    return mul;

}
```

```
//25 Faibnocci
#include <stdio.h>
int main() {

  int i, n;

  // initialize first and second terms
  int t1 = 0, t2 = 1;

  // initialize the next term (3rd term)
  int nextTerm = t1 + t2;

  // get no. of terms from user
  printf("Enter the number of terms: ");
  scanf("%d", &n);

  // print the first two terms t1 and t2
  printf("Fibonacci Series: %d, %d, ", t1, t2);

  // print 3rd to nth terms
  for (i = 3; i <= n; ++i) {
    printf("%d, ", nextTerm);
    t1 = t2;
    t2 = nextTerm;
    nextTerm = t1 + t2;
  }

  return 0;
}
OUTPUT:
Enter the number of terms: 10
Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
//26(A )Finding armstrong number
/**
 * C program to print Armstrong numbers from 1 to n
 */
#include <stdio.h>
#include <math.h>

int main()
{
    int num, lastDigit, digits, sum, i, end;

    /* Input upper limit from user */
    printf("Enter upper limit: ");
    scanf("%d", &end);

    printf("Armstrong number between 1 to %d are: \n", end);

    for(i=1; i<=end; i++)
    {
        sum = 0;

        /* Copy the value of num for processing */
        num = i;

        /* Find total digits in num */
        digits = (int) log10(num) + 1;

        /* Calculate sum of power of digits */
        while(num > 0)
        {
            /* Extract last digit */
            lastDigit = num % 10;

            // Find sum of power of digits
            // Use ceil() function to overcome any rounding errors by
pow()
            sum = sum + ceil(pow(lastDigit, digits));

            /* Remove the last digit */
            num = num / 10;
        }

        /* Check for Armstrong number */
        if(i == sum)
        {
            printf("%d, ", i);
        }

    }

    return 0;
}
//Checking Armstrong number
#include <stdio.h>
```

```
int main() {
    int num, originalNum, remainder, result = 0;
    printf("Enter a three-digit integer: ");
    scanf("%d", &num);
    originalNum = num;

    while (originalNum != 0) {
       // remainder contains the last digit
        remainder = originalNum % 10;

       result += remainder * remainder * remainder;

       // removing last digit from the orignal number
       originalNum /= 10;
    }

    if (result == num)
        printf("%d is an Armstrong number.", num);
    else
        printf("%d is not an Armstrong number.", num);

    return 0;
}
```

OUTPUT:

```
Enter a three-digit integer: 371
371 is an Armstrong number.
```

```
//27(A)Checking number is perfect or not
/*C program to check whether the given number is the Perfect number*/
#include<stdio.h>
#include<conio.h>
void main()
{
// declare and initialize the variables
int num, rem, sum = 0, i;
// take an input from the user.
printf("Enter a number\n");
scanf("%d", &num);
// find all divisors and add them
for(i = 1; i < num; i++)
                     {
                              rem = num % i;
                             if (rem == 0)
                                        {
                                               sum = sum + i;
                                         }
                        }
if (sum == num)
                      printf(" %d is a Perfect Number");
           else
                      printf("\n %d is not a Perfect Number");
getch();
}
```

```
Output:
Enter a number
28
Entered number is perfect
```

```
//27(B)Printing perfect number b/w 1 to n
/**
 * C program to print all Perfect numbers between 1 to n
 */

#include <stdio.h>

int main()
{
    int i, j, end, sum;

    /* Input upper limit to print perfect number */
    printf("Enter upper limit: ");
    scanf("%d", &end);

    printf("All Perfect numbers between 1 to %d:\n", end);

    /* Iterate from 1 to end */
    for(i=1; i<=end; i++)
    {
        sum = 0;

        /* Check whether the current number i is Perfect number or not */
        for(j=1; j<i; j++)
        {
            if(i % j == 0)
            {
                sum += j;
            }
        }

        /* If the current number i is Perfect number */
        if(sum == i)
        {
            printf("%d, ", i);
        }
    }

    return 0;
}
```

```c
//28 power of any number
/**
 * C program to find power of any number using for loop
 */

#include <stdio.h>

int main()
{
    int base, exponent;
    long long power = 1;
    int i;

    /* Input base and exponent from user */
    printf("Enter base: ");
    scanf("%d", &base);
    printf("Enter exponent: ");
    scanf("%d", &exponent);

    /* Multiply base, exponent times*/
    for(i=1; i<=exponent; i++)
    {
        power = power * base;
    }

    printf("%d ^ %d = %lld", base, exponent, power);

    return 0;
}
```

```
//Ascii value of all character
#include < stdio.h >

int main() {
    char ch;

    // assigning a letter to ch
    printf("Enter the Character: ");
    scanf("%c", & ch);

    // displaying the ASCII value of the letter stored in ch
    printf("\nThe ASCII Value of %c is %d", ch, ch);

    return 0;
}
```

Input:

```
Enter the Character: z
```

Output:

```
The ASCII Value of z is 122
```

```
//30 pascal trangle up to n row
/**
 * C program to print Pascal triangle up to n rows
 */

#include <stdio.h>

/* Function definition */
long long fact(int n);

int main()
{
    int n, k, num, i;
    long long term;

    /* Input number of rows */
    printf("Enter number of rows : ");
    scanf("%d", &num);

    for(n=0; n<num; n++)
    {
        /* Prints 3 spaces */
        for(i=n; i<=num; i++)
            printf("%3c", ' ');

        /* Generate term for current row */
        for(k=0; k<=n; k++)
        {
            term = fact(n) / (fact(k) * fact(n-k));

            printf("%6lld", term);
        }

        printf("\n");
    }

    return 0;
}

/**
 * Function to calculate factorial
 */
long long fact(int n)
{
    long long factorial = 1ll;
    while(n>=1)
    {
        factorial *= n;
        n--;
    }

    return factorial;
}
OUTPUT:
Enter number of rows : 10
                         1
                       1   1
                     1   2   1
                   1   3   3   1
                 1   4   6   4   1
               1   5  10  10   5   1
             1   6  15  20  15   6   1
           1   7  21  35  35  21   7   1
         1   8  28  56  70  56  28   8   1
```

```
1   9  36  84 126 126  84  36   9   1
```

```
//31 Array sum
#include<stdio.h>

int main()
{
    //let's assume the maximum array size as 100.
    //initialize sum as 0. Otherwise, it will take some garbage value.
    int arr[100], size, i, sum = 0;

    //Get size input from user
    printf("Enter array size\n");
    scanf("%d",&size);

    //Get all elements using for loop and store it in array
    printf("Enter array elements\n");
    for(i = 0; i < size; i++)
          scanf("%d",&arr[i]);

    //add all elements to the variable sum.
    for(i = 0; i < size; i++)
          sum = sum + arr[i]; // same as sum += arr[i];

    //print the result
    printf("Sum of the array = %d\n",sum);

    return 0;
}
```

```
OUTPUT:
Input
array size = 4

Elements = {12, -1, 0, 8}

Output
19
```

```
//32 Copy one array to another
#include <stdio.h>

int main()
{
    //Initialize array
    int arr1[] = {1, 2, 3, 4, 5};

    //Calculate length of array arr1
    int length = sizeof(arr1)/sizeof(arr1[0]);

    //Create another array arr2 with the size of arr1.

    int arr2[length];

    //Copying all elements of one array into another
    for (int i = 0; i < length; i++) {
        arr2[i] = arr1[i];
    }

    //Displaying elements of array arr1
    printf("Elements of original array: \n");
    for (int i = 0; i < length; i++) {
        printf("%d ", arr1[i]);
    }

    printf("\n");

    //Displaying elements of array arr2
    printf("Elements of new array: \n");
    for (int i = 0; i < length; i++) {
        printf("%d ", arr2[i]);
    }
    return 0;
}
```

Output:

```
Elements of original array
1 2 3 4 5
Elements of new array:
1 2 3 4 5
```

```
\\33 Array insertion
\* C Program to to insert an element at given position in an array *\

# include < stdio.h >
int  main( )
{
int  a[20], i, n, ele, pos ;
printf(" Enter the Numbers of elements: ") ;
scanf("%d ",& n) ;
printf("\n Enter the elements of array : \n") ;
for (  i = 1 ; i < = n ; i++)
scanf("%d ",& a[i]) ;
printf("\n Array enter by user are :\n") ;
for (  i = 1 ; i < = n ; i++)
printf("%d \t",a[i]) ;

printf("\n Enter the position you want to enter :") ;
scanf("%d ",& pos) ;
printf("\n Enter the element you want to enter :") ;
scanf("%d ",& ele) ;

for (  i = 1 ; i < = n ; i++)
{
if ( i > pos )
a[i] = a[i-1] ;
else
{
if ( i == pos )
a[i] = ele ;
else
a[i] = a[i] ;
}
}
printf("\n Array After Inserting element :\n") ;
for (  i = 1 ; i < = n ; i++)
pcanf("%d \t",a[i]) ;
return ( 0 ) ;
}
```

OUTPUT:

```
Enter the number of the position 5
Enter array elements 12
32
42
21
23
Enter position that you want to enter 55
Array after inserting elements 12  32  42  55  21  23
```

```
//34 Deletion of the array
/*
 * C program to delete an element from array at specified position
 */

#include <stdio.h>
#define MAX_SIZE 100

int main()
{
    int arr[MAX_SIZE];
    int i, size, pos;

    /* Input size and element in array */
    printf("Enter size of the array : ");
    scanf("%d", &size);
    printf("Enter elements in array : ");
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }

    /* Input element position to delete */
    printf("Enter the element position to delete : ");
    scanf("%d", &pos);


    /* Invalid delete position */
    if(pos < 0 || pos > size)
    {
        printf("Invalid position! Please enter position between 1 to %d", size);
    }
    else
    {
        /* Copy next element value to current element */
        for(i=pos-1; i<size-1; i++)
        {
            arr[i] = arr[i + 1];
        }

        /* Decrement array size by 1 */
        size--;

        /* Print array after deletion */
        printf("\nElements of array after delete are : ");
        for(i=0; i<size; i++)
        {
            printf("%d\t", arr[i]);
        }
    }

    return 0;
}
```

```
 Output
Enter size of the array : 5
Enter elements in array : 10 20 30 40 50
Enter the element position to delete : 2

Elements of array after delete are : 10      30      40      50
```

```
//Linear searching
/*
* C Program to search any element or number in an array
*/
#include <stdio.h>
int main(){
    int inputArray[100], elementCount, counter, num;

    printf("Enter Number of Elements in Array\n");
    scanf("%d", &elementCount);
    printf("Enter %d numbers \n", elementCount);

    /* Read array elements */
    for(counter = 0; counter < elementCount; counter++){
        scanf("%d", &inputArray[counter]);
    }

    printf("Enter a number to serach in Array\n");
    scanf("%d", &num);

    /* search num in inputArray from index 0 to elementCount-1 */
    for(counter = 0; counter < elementCount; counter++){
        if(inputArray[counter] == num){
            printf("Number %d found at index %d\n", num, counter);
            break;
        }
    }

    if(counter == elementCount){
     printf("Number %d Not Present in Input Array\n", num);
    }
    return 0;
}
```

Output

```
Enter Number of Elements in Array
6
Enter 6 numbers
7 2 9 4 1 6
Enter a number to serach in Array
4
Number 4 found at index 3
```

```
//36 (A)Second largest
/**
 * C program to find second largest number in an array
 */

#include <stdio.h>
#include <limits.h> // For INT_MIN

#define MAX_SIZE 1000     // Maximum array size

int main()
{
    int arr[MAX_SIZE], size, i;
    int max1, max2;

    /* Input size of the array */
    printf("Enter size of the array (1-1000): ");
    scanf("%d", &size);

    /* Input array elements */
    printf("Enter elements in the array: ");
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }

    max1 = max2 = INT_MIN;


    /*
     * Check for first largest and second
     */
    for(i=0; i<size; i++)
    {
        if(arr[i] > max1)
        {
            /*
             * If current element of the array is first largest
             * then make current max as second max
             * and then max as current array element
             */
            max2 = max1;
            max1 = arr[i];
        }
        else if(arr[i] > max2 && arr[i] < max1)
        {
            /*
             * If current array element is less than first largest
             * but is greater than second largest then make it
             * second largest
             */
            max2 = arr[i];
        }
    }

    printf("First largest = %d\n", max1);
    printf("Second largest = %d", max2);

    return 0;
}
```

Output

```
Enter size of the array (1-1000): 10
Enter elements in the array: -7 2 3 8 6 6 75 38 3 2
First largest = 75
```

```
Second largest = 38
```

```
//Sorting using bubble sort
#include <stdio.h>

int main(){

    int arr[50], num, x, y, temp;

    printf("Please Enter the Number of Elements you want in the array: ");

    scanf("%d", &num);

    printf("Please Enter the Value of Elements: ");

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

        scanf("%d", &arr[x]);

    for(x = 0; x < num - 1; x++){

        for(y = 0; y < num - x - 1; y++){

            if(arr[y] > arr[y + 1]){

                temp = arr[y];

                arr[y] = arr[y + 1];

                arr[y + 1] = temp;

            }

        }

    }

    printf("Array after implementing bubble sort: ");

    for(x = 0; x < num; x++){

        printf("%d  ", arr[x]);

    }

    return 0;

}
```

```
Output:
Enter number of elements in array
```

```
/**
 * C program to sort elements of array in ascending order
 */
//bubble sorting
#include <stdio.h>
    // Maximum array size

int main()
{
    int arr[100];
    int size;
    int i, j, temp;

    /* Input size of array */
    printf("Enter size of array: ");
    scanf("%d", &size);

    /* Input elements in array */
    printf("Enter elements in array: ");
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }

    for(i=0; i<size; i++)
    {
        /*
         * Place currently selected element array[i]
         * to its correct place.
         */
        for(j=i+1; j<size; j++)
        {
            /*
             * Swap if currently selected array element
             * is not at its correct position.
             */
            if(arr[i] > arr[j])
            {
                temp     = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }

    /* Print the sorted array */
    printf("\nElements of array in ascending order: ");
    for(i=0; i<size; i++)
    {
        printf("%d\t", arr[i]);
    }

    return 0;
}
```

```
//37 counting total number of duplicate element in array
/**
 * C program to count total number of duplicate elements in an array
 */

#include <stdio.h>
  // Maximum array size

int main()
{
    int arr[100];
    int i, j, size, count = 0;

    /* Input size of array */
    printf("Enter size of the array : ");
    scanf("%d", &size);

    /* Input elements in array */
    printf("Enter elements in array : ");
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }

    /*
     * Find all duplicate elements in array
     */
    for(i=0; i<size; i++)
    {
        for(j=i+1; j<size; j++)
        {
            /* If duplicate found then increment count by 1 */
            if(arr[i] == arr[j])
            {
                count++;
                break;
            }
        }
    }

    printf("\nTotal number of duplicate elements found in array = %d", count);

    return 0;
}
Output
Enter size of the array : 10
Enter elements in array : 1 10 20 1 25 1 10 30 25 1

Total number of duplicate elements found in array = 5
```

```c
/**
 * C program to perform scalar matrix multiplication
 */

#include <stdio.h>

#define SIZE 3 // Maximum size of the array

int main()
{
    int A[SIZE][SIZE];
    int num, row, col;

    /* Input elements in matrix from user */
    printf("Enter elements in matrix of size %dx%d: \n", SIZE, SIZE);
    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {
            scanf("%d", &A[row][col]);
        }
    }

    /* Input multiplier from user */
    printf("Enter any number to multiply with matrix A: ");
    scanf("%d", &num);

    /* Perform scalar multiplication of matrix */
    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {
            /* (cAij) = c . Aij */
            A[row][col] = num * A[row][col];
        }
    }

    /* Print result of scalar multiplication of matrix */
    printf("\nResultant matrix c.A = \n");
    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {
            printf("%d ", A[row][col]);
        }
        printf("\n");
    }

    return 0;
}
```

Output

```
Enter elements in matrix of size 3x3:
1 2 3
4 5 6
7 8 9
Enter any number to multiply with matrix A: 2

Resultant matrix c.A =
2  4  6
8 10 12
14 16 18
```

```
//39 sum of main diagonal
/**
 * C program to find sum of main diagonal elements of a matrix
 */

#include <stdio.h>

#define SIZE 3 // Matrix size

int main()
{
    int A[SIZE][SIZE];
    int row, col, sum = 0;

    /* Input elements in matrix from user */
    printf("Enter elements in matrix of size %dx%d: \n", SIZE, SIZE);
    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {
            scanf("%d", &A[row][col]);
        }
    }

    /* Find sum of main diagonal elements */
    for(row=0; row<SIZE; row++)
    {
        sum = sum + A[row][row];
    }

    printf("\nSum of main diagonal elements = %d", sum);

    return 0;
}
```

Output

```
Enter elements in matrix of size 3x3:
1 2 3
4 5 6
7 8 9

Sum of main diagonal elements = 15
```

```c
//40(a)Sparse matrix
/**
 * C program to check sparse matrix
 */

#include <stdio.h>
#define SIZE 3

int main()
{
    int A[SIZE][SIZE];
    int row, col, total=0;

    /* Input elements in matrix from user */
    printf("Enter elements in matrix of size 3x3: \n");
    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {
            scanf("%d", &A[row][col]);
        }
    }

    /* Count total number of zero elements in the matrix */
    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {
            /* If the current element is zero */
            if(A[row][col] == 0)
            {
                total++;
            }
        }
    }

    if(total >= (row * col)/2)
    {
        printf("\nThe given matrix is a Sparse matrix.");
    }
    else
    {
        printf("\nThe given matrix is not Sparse matrix.");
    }

    return 0;
}
```

Output

```
Enter elements in matrix of size 3x3:
1 0 0
4 5 0
0 0 0

The given matrix is a Sparse matrix.
```

```
//40(b)Transpose of matrix
#include <stdio.h>
int main() {
  int a[10][10], transpose[10][10], r, c;
  printf("Enter rows and columns: ");
  scanf("%d %d", &r, &c);

  // asssigning elements to the matrix
  printf("\nEnter matrix elements:\n");
  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    printf("Enter element a%d%d: ", i + 1, j + 1);
    scanf("%d", &a[i][j]);
  }

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

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

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

Output

```
Enter rows and columns: 2
3

Enter matrix elements:
Enter element a11: 1
Enter element a12: 4
Enter element a13: 0
Enter element a21: -5
Enter element a22: 2
Enter element a23: 7

Entered matrix:
```

```
1  4  0
-5  2  7

Transpose of the matrix:
1  -5
4  2
0  7
```

```
//41 Identity matrix checking
/*
* C Program to check if a given matrix is an identity matrix
*/
#include <stdio.h>

int main (void)
{
        int a[10][10];
        int i = 0, j = 0, row = 0, col = 0;

        printf ("Enter the order of the matrix (mxn):\n");
        printf ("where m = number of rows; and\n");
        printf ("      n = number of columns\n");
        scanf ("%d %d", &row, &col);

        int flag = 0;

        printf ("Enter the elements of the matrix\n");
        for (i = 0; i < row; i++)
        {
                for (j = 0; j < col; j++)
                {
                        scanf ("%d", &a[i][j]);
                }
        }

        for (i = 0; i < row; i++)
        {
                for (j = 0; j < col; j++)
                {
                        if (i == j && a[i][j] != 1)
                        {
                                flag = -1;
                                break;
                        }
                        else if (i != j && a[i][j] != 0)
                        {
                                flag = -1;
                                break;
                        }
                }
        }

        if (flag == 0)
        {
                printf ("It is a IDENTITY MATRIX\n");
        }
        else
        {
                printf ("It is NOT an identity matrix\n");
        }

        return 0;
}
OUTPUT;
```

```
Enter the order of the matrix (mxn):
where m = number of rows; and
      n = number of columns
3 3
Enter the elements of the matrix
1 0 0
0 1 0
0 0 1
It is a IDENTITY MATRIX

Enter the order of the matrix (mxn):
where m = number of rows; and
      n = number of columns
3 3
Enter the elements of the matrix
1 2 3
4 5 6
5 8 7
It is NOT an identity matrix
```

```
//42\* C Program to to merge two sorted array in ascending order *\

# include < stdio.h >
int  main( )
{
int  a[50], b[25], i, j, k=1, s, m, n, temp ;
printf(" Enter the number of element in first array : ") ;
scanf("%d ",& m) ;
printf("\n Enter the element of first array in ascending order : \n") ;
for (  i = 1 ; i < = m ; i++)
scanf("%d ",& a[i]) ;

printf(" Enter the number of element in second array : ") ;
scanf("%d ",& n) ;
printf("\n Enter the element of second array in ascending order : \n") ;
for (  i = 1 ; i < = n ; i++)
scanf("%d ",& b[i]) ;
s = m + n ;

for (  i = m+1 ; i <= s ; i++)
{
a[i] = b[k] ;
for (  j = 1 ; j <= s ; j++)
{
if (  a[j] >= a[i] )
{
temp = a[i] ;
a[j] = a[i] ;
a[i] = temp ;
}
}
k = k+1 ;
}
printf("\n Array after merging :\n") ;
for (  i = 1 ; i <= s ; i++)
pcanf("%d \t",a[i]) ;
return ( 0 ) ;
}
Output of Program:
Input: arr1[] = { 1, 3, 4, 5}, arr2[] = {2, 4, 6, 8}
Output: arr3[] = {1, 2, 3, 4, 4, 5, 6, 8}

Input: arr1[] = { 5, 8, 9}, arr2[] = {4, 7, 8}
Output: arr3[] = {4, 5, 7, 8, 8, 9}
```

```
//43(a)COMPARE
#include <stdio.h>
#include <string.h>

int main() {

  char s1[20] = "ScalerAcademy"; // string1
  char s2[20] = "ScalerAcademy.COM"; // string2
  // comparing both the strings
  if (strcmp(s1, s2) == 0) {
    printf("string 1 and string 2 are equal");
  } else {
    printf("string 1 and 2 are different");
  }
}
```

Output

```
string 1 and 2 are different
```

```
//43(b)CONCATENATION
#include <stdio.h>
#include <string.h>

int main() {

  char string1[10] = "Hello";
  char string2[10] = "World";
  strcat(string1, string2);
  printf("Output string after concatenation: %s", string1);

}
```

Output

```
Output string after concatenation: HelloWorld
```

```
//43(c)COPY OF STRING
#include <stdio.h>
#include <string.h>

int main() {

  char s1[35] = "string 1"; // string1
  char s2[35] = "I'll be copied to string 1."; // string2
  strcpy(s1, s2); // copying string2 to string1
  printf("String s1 is: %s", s1); // printing string1

}
```

Output

```
String s1 is: I'll be copied to string 1.
```

```
//43(d)LENGTH OF STRING
#include <stdio.h>
#include <string.h>

int main() {

  char string1[20] = "ScalerAcademy";
  printf("Length of string string1: %ld", strlen(string1));
  return 0;
}
```

Output

```
Length of string string1: 13
```

```c
//REVERSING IN STRING
#include <stdio.h>
int main()
{
  char str[1000], rev[1000];
  int i, j, count = 0;
  scanf("%s", str);
  printf("\nString Before Reverse: %s", str);
  //finding the length of the string
  while (str[count] != '\0')
  {
    count++;
  }
  j = count - 1;

  //reversing the string by swapping
  for (i = 0; i < count; i++)
  {
    rev[i] = str[j];
    j--;
  }

  printf("\nString After Reverse: %s", rev);
}
```

```
Hello

String Before Reverse: Hello
String After Reverse: olleH
```

```c
//UPPERCASE AND LOWERCASE IN STRING
#include <stdio.h>
#include <conio.h>
int main ()
{
    char upr, lwr; // declare variables
    int ascii;

    // convert in lower case
    printf (" Enter the Upper Case Character: ");
    scanf (" %c", &upr);
    ascii = upr + 32;
    printf (" %c character in Lower case is: %c", upr, ascii);

    // convert in upper case
    printf (" \n Enter the Lower Case Character: ");
    scanf (" %c", &lwr);
    ascii = lwr - 32;
    printf (" %c character in the Upper case is: %c", lwr, ascii);

    return 0;
}
```

Output

```
Enter the Upper Case Character: A
 A character in Lower case is: a
 Enter the Lower Case Character: z
 z character in the Upper case is: Z
```

```
//44 palindrome in string
#include<stdio.h>

int main()
{
    char string[40];
    int length=0, flag=1,i;

    printf("Enter string:\n");
    gets(string);

    for(i=0;string[i]!='\0';i++)
    {
        length++;
    }

    for(i=0;i< length/2;i++)
    {
        if( string[i] != string[length-1-i] )
        {
            flag=0;
            break;
        }
    }

    if(flag==1)
    {
        printf("PALINDROME");
    }
    else
    {
        printf("NOT PALINDROME");
    }
    return 0;
}
```

Output

```
Run 1:
------------
Enter String:
madam
PALINDROME

Run 2:
------------
Enter String:
step on no pets
PALINDROME

Run 3:
------------
Enter String:
madam
codesansar
```

```
NOT PALINDROME
```

```
//45 frequency of each character in string
#include <stdio.h>
#include <string.h>

int main()
{
    char s[1000];
    int  i,j,k,count=0,n;

    printf("Enter  the string : ");
    gets(s);

    for(j=0;s[j];j++);
       n=j;

      printf(" frequency count character in string:\n");

    for(i=0;i<n;i++)
    {
      count=1;
      if(s[i])
      {

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

              if(s[i]==s[j])
          {
                 count++;
                 s[j]='\0';
            }
            }
            printf(" '%c' = %d \n",s[i],count);



       }


      }


    return 0;
}
Output:

 Enter  the string: hello world
 frequency count character in string:
 'h' = 1
 'e' = 1
 'l' = 3
 'o' = 2
 ' ' = 1
 'w' = 1
```

```
'r' = 1
'd' = 1

Enter  the string : abcd abc ab a
frequency count character in string:
'a' = 4
'b' = 3
'c' = 2
'd' = 1
' ' = 3
```

```
//46 FINDING DIAMETER,CIRCUMFERENCE AND AREA OF CIRCLE
/**
 * C program to find diameter, circumference and area of a circle using
functions
 */

#include <stdio.h>
#include <math.h> // Used for constant PI referred as M_PI


/* Function declaration */
double getDiameter(double radius);
double getCircumference(double radius);
double getArea(double radius);


int main()
{
    float radius, dia, circ, area;

    /* Input radius of circle from user */
    printf("Enter radius of circle: ");
    scanf("%f", &radius);

    dia  = getDiameter(radius);       // Call getDiameter function
    circ = getCircumference(radius);  // Call getCircumference function
    area = getArea(radius);           // Call getArea function

    printf("Diameter of the circle = %.2f units\n", dia);
    printf("Circumference of the circle = %.2f units\n", circ);
    printf("Area of the circle = %.2f sq. units", area);

    return 0;
}


/**
 * Calculate diameter of circle whose radius is given
 */
double getDiameter(double radius)
{
    return (2 * radius);
}


/**
 * Calculate circumference of circle whose radius is given
 */
double getCircumference(double radius)
{
    return (2 * M_PI * radius); // M_PI = PI = 3.14 ...
}
```

```
/**
 * Find area of circle whose radius is given
 */
double getArea(double radius)
{
    return (M_PI * radius * radius); // M_PI = PI = 3.14 ...
}
```

Output

```
Enter radius of the circle: 10
Diameter of the circle = 20.00 units
Circumference of the circle = 62.83 units
Area of the circle = 314.16 sq. units
```

```c
//47
/**
 * C program to check prime, armstrong and perfect numbers using
functions
 */

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


/* Function declarations */
int isPrime(int num);
int isArmstrong(int num);
int isPerfect(int num);


int main()
{
    int num;

    printf("Enter any number: ");
    scanf("%d", &num);

    // Call isPrime() functions
    if(isPrime(num))
    {
        printf("%d is Prime number.\n", num);
    }
    else
    {
        printf("%d is not Prime number.\n", num);
    }

    // Call isArmstrong() function
    if(isArmstrong(num))
    {
        printf("%d is Armstrong number.\n", num);
    }
    else
    {
        printf("%d is not Armstrong number.\n", num);
    }

    // Call isPerfect() function
    if(isPerfect(num))
    {
        printf("%d is Perfect number.\n", num);
    }
    else
    {
        printf("%d is not Perfect number.\n", num);
    }

    return 0;
```

```
}



/**
 * Check whether a number is prime or not.
 * Returns 1 if the number is prime otherwise 0.
 */
int isPrime(int num)
{
    int i;

    for(i=2; i<=num/2; i++)
    {
        /*
         * If the number is divisible by any number
         * other than 1 and self then it is not prime
         */
        if(num%i == 0)
        {
            return 0;
        }
    }

    return 1;
}



/**
 * Check whether a number is Armstrong number or not.
 * Returns 1 if the number is Armstrong number otherwise 0.
 */
int isArmstrong(int num)
{
    int lastDigit, sum, originalNum, digits;
    sum = 0;

    originalNum = num;

    /* Find total digits in num */
    digits = (int) log10(num) + 1;

    /*
     * Calculate sum of power of digits
     */
    while(num > 0)
    {
        // Extract the last digit
        lastDigit = num % 10;

        // Compute sum of power of last digit
        sum = sum + round(pow(lastDigit, digits));
```

```c
        // Remove the last digit
        num = num / 10;
    }

    return (originalNum == sum);
}



/**
 * Check whether the number is perfect number or not.
 * Returns 1 if the number is perfect otherwise 0.
 */
int isPerfect(int num)
{
    int i, sum, n;
    sum = 0;
    n = num;

    for(i=1; i<n; i++)
    {
        /* If i is a divisor of num */
        if(n%i == 0)
        {
            sum += i;
        }
    }

    return (num == sum);
}
```

Output

```
Enter any number: 11
11 is Prime number.
11 is not Armstrong number.
11 is not Perfect number.
```

```
//48 Adding two number using pointers
#include <stdio.h>
int main()
{
   int first, second, *p, *q, sum;

   printf("Enter two integers to add\n");
   scanf("%d%d", &first, &second);

   p = &first;
   q = &second;

   sum = *p + *q;

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

   return 0;
}
OUTPUT;
Enter two integer to add5
6
Sum of entered number is;11
```

```
//49(a) Swapping 2 numbers using Call by Value
#include <stdio.h>


void swap(int, int);

int main()
{
   int x, y;

   printf("Enter the value of x and y\n");
   scanf("%d%d",&x,&y);

   printf("Before Swapping\nx = %d\ny = %d\n", x, y);

   swap(x, y);

   printf("After Swapping\nx = %d\ny = %d\n", x, y);

   return 0;
}

void swap(int a, int b)
{
   int temp;

   temp = b;
   b = a;
   a = temp;
    printf("Values of a and b is %d  %d\n",a,b);
}
```

Output:

```
Enter the value of x and y
Before Swapping
x = 10
y = 5
Values of a and b is 5 10
After Swapping
x = 10
y = 5
```

```
//49(b)Swaping two number call by reference
#include <stdio.h>

void swap(int*, int*);

int main()
{
   int x, y;
   printf("Enter the value of x and y\n");
   scanf("%d%d",&x,&y);

   printf("Before Swapping\nx = %d\ny = %d\n", x, y);

   swap(&x, &y);

   printf("After Swapping\nx = %d\ny = %d\n", x, y);

   return 0;
}

void swap(int *a, int *b)
{
   int temp;

   temp = *b;
   *b = *a;
   *a = temp;
}
```

```
//50(a) Copying an array to another array using pointer
/**
 * C program to copy an array to another array using pointers
 */

#include <stdio.h>

#define MAX_SIZE 100 // Maximum array size


/* Function declaration to print array */
void printArray(int arr[], int size);


int main()
{
    int source_arr[MAX_SIZE], dest_arr[MAX_SIZE];
    int size, i;

    int *source_ptr = source_arr;   // Pointer to source_arr
    int *dest_ptr   = dest_arr;     // Pointer to dest_arr

    int *end_ptr;


    /*
     * Input size and elements in source array
     */
    printf("Enter size of array: ");
    scanf("%d", &size);
    printf("Enter elements in array: ");
    for (i = 0; i < size; i++)
    {
        scanf("%d", (source_ptr + i));
    }


    // Pointer to last element of source_arr
    end_ptr = &source_arr[size - 1];


    /* Print source and destination array before copying */
    printf("\nSource array before copying: ");
    printArray(source_arr, size);

    printf("\nDestination array before copying: ");
    printArray(dest_arr, size);



    /*
     * Run loop till source_ptr exists in source_arr
     * memory range.
     */
    while(source_ptr <= end_ptr)
    {
        *dest_ptr = *source_ptr;

        // Increment source_ptr and dest_ptr
        source_ptr++;
        dest_ptr++;
    }


    /* Print source and destination array after copying */
    printf("\n\nSource array after copying: ");
    printArray(source_arr, size);

    printf("\nDestination array after copying: ");
    printArray(dest_arr, size);


    return 0;
}


/**
 * Function to print array elements.
 *
 * @arr     Integer array to print.
 * @size    Size of array.
 */
void printArray(int *arr, int size)
{
    int i;

    for (i = 0; i < size; i++)
    {
        printf("%d, ", *(arr + i));
    }
}
```

```
while(source_ptr <= end_ptr)
    *(dest_ptr++) = *(source_ptr++);
```

Output

```
Enter size of array: 10
Enter elements in array: 10 -1 100 90 87 0 15 10 20 30

Source array before copying: 10, -1, 100, 90, 87, 0, 15, 10, 20, 30,
Destination array before copying: 0, 0, 127, 127, 0, 1, 0, 16777472, 0,
0,

Source array after copying: 10, -1, 100, 90, 87, 0, 15, 10, 20, 30,
Destination array after copying: 10, -1, 100, 90, 87, 0, 15, 10, 20, 0,
```

```c
//50(b) Reverse an array using pointer
/**
 * C program to reverse an array using pointers
 */

#include <stdio.h>

#define MAX_SIZE 100

/* Function declaration */
void printArr(int *arr, int size);


int main()
{
    int arr[MAX_SIZE];
    int size;
    int *left = arr;  // Pointer to arr[0]
    int *right;


    // Input size of array
    printf("Enter size of array: ");
    scanf("%d", &size);

    right = &arr[size - 1];  // Pointer to arr[size - 1]

    /*
     * Input elements in array
     */
    printf("Enter elements in array: ");
    while(left <= right)
    {
        scanf("%d", left++);
    }


    printf("\nArray before reverse: ");
    printArr(arr, size);


    // Make sure that left points to arr[0]
    left = arr;


    // Loop to reverse array
    while(left < right)
    {
        /*
         * Swap element from left of array to right of array.
         */
        *left    ^= *right;
        *right   ^= *left;
        *left    ^= *right;

        // Increment left array pointer and decrement right array pointer
        left++;
        right--;
    }


    printf("\nArray after reverse: ");
    printArr(arr, size);


    return 0;
}



/**
 * Function to print array using pointer.
 *
 * @arr     Pointer to array.
 * @size    Size of the array.
 */
void printArr(int * arr, int size)
{
    // Pointer to arr[size - 1]
    int * arrEnd = (arr + size - 1);

    /* Loop till last array element */
    while(arr <= arrEnd)
    {
        printf("%d, ", *arr);

        // Move pointer to next array element.
        arr++;
    }
}
```

Output

```
Enter size of array: 10
Enter elements in array: 10 20 30 40 50 60 70 80 90 100

Array before reverse: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,
Array after reverse: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10
```

Pattern Printing in C

Output :

```
 * * * * *
 * * * * *
 * * * * *
 * * * * *
 * * * * *_
```

Program :

```
/*  Program to print pyramid pattern in C : Pattern 1

#include <stdio.h>
#include <conio.h>
void main()
{
        int i,j;
        clrscr();
        for(i=0; i<5; i++)
        {
                for(j=0; j<5; j++)
                {
                        printf(" * ");
                }
                printf("\n");
        }
        getch();
}
```

Output :

```
 *
 * *
 * * *
 * * * *
 * * * * *_
```

Program :

```
/*  Program to print pyramid pattern in C : Pattern 2

#include <stdio.h>
#include <conio.h>
void main()
{
        int i,j;
        clrscr();
        for(i=0; i<5; i++)
        {
                for(j=0; j<=i; j++)
                {
                        printf(" * ");
                }
                printf("\n");
        }
        getch();
}
```

Output :

```
         *
       * *
     * * *
   * * * *
 * * * * *_
```

Program :

```
/*  Program to print pyramid pattern in C : Pattern 3

#include <stdio.h>
#include <conio.h>
void main()
{
        int i,j,k;
        clrscr();
        for(i=1; i<=5; i++)
        {
                for(j=5; j>=i; j--)
                {
                        printf(" ");
                }
                for(k=1; k<=i; k++)
                {
                        printf("*");
                }
                printf("\n");
        }
        getch();
}
```

Output :

```
 * * * * *
   * * * *
     * * *
       * *
         *_
```

Program :

```
/*  Program to print pyramid pattern in C : Pattern 4

#include <stdio.h>
#include <conio.h>
void main()
{
       int i,j,k,samp=1;
       clrscr();
       for(i=5; i>=1; i--)
       {
              for(k=samp; k>=0; k--)
              {
                     printf(" "); // only 1 space
              }
              for(j=i; j>=1; j--)
              {
                     printf("*");
              }
              samp = samp + 1;
              printf("\n");
       }
       getch();
}
```

or

```
#include <stdio.h>
#include <conio.h>
void main()
{
       int i,j,k;
       clrscr();
       for(i=1; i<=5; i++)
       {
              for(j=5; j>=i; j--)
              {
                     printf("*");
              }
              for(k=1; k<=i; k++)
              {
                     printf(" ");
              }
              printf("\n");
       }
       getch();
}
```

Output :

```
 * * * * *
 * * * *
 * * *
 * *
 *_
```

Program :

```
/*  Program to print pyramid pattern in C : Pattern 5


#include <stdio.h>
#include <conio.h>
void main()
{
	int i,j;
	clrscr();
	for(i=5; i>=1; i--)
	{
		for(j=1; j<=i; j++)
		{
			printf(" * ");
		}
		printf("\n");
	}
	getch();
}
```

Output :

```
     *
    * *
   * * *
  * * * *
 * * * * *_
```

Program :

```
/*  Program to print pyramid pattern in C : Pattern 6


#include <stdio.h>
#include <conio.h>
void main()
{
	int i,j,k,t=0;
	clrscr();
	for(i=1; i<=5; i++)
	{
		for(k=t; k<5; k++)
		{
			printf(" ");
		}
		for(j=0; j< i; j++)
		{
			printf(" * ");
			t = t + 1;
		}
		printf("\n");
	}
	getch();
}
```

Output :

```
        *
      * *
    * * *
  * * * *
* * * * *
  * * * *
    * * *
      * *
        *_
```

Program :

```
/*  Program to print pyramid pattern in C : Pattern 7

Creation Date : 01:13 AM 22/11/2010

Author : www.technoexam.com [Technowell, Sangli] */

#include <stdio.h>
#include <conio.h>
void main()
{
       int i,j,k,samp=1;
       clrscr();
       for(i=1; i<=5; i++)
       {
              for(k=samp; k<=5; k++)
              {
                     printf(" ");
              }
              for(j=0; j< i; j++)
              {
                     printf("*");
              }
              samp = samp + 1;
              printf("\n");
       }
       samp = 1;
       for(i=4; i>=1; i--)
       {
              for(k=samp; k>=0; k--)
              {
                     printf(" ");
              }
              for(j=i; j>=1; j--)
              {
                            printf("*");
              }
              samp = samp + 1;
              printf("\n");
       }
       getch();
}
```

Output :

```
Enter number of rows: 5

1
2 3
4 5 6
7 8 9 10
11 12 13 14 15_
```

Program :

```
/*  Program to print pyramid pattern in C : Pattern 8

Creation Date : 02:39 PM 01/10/2011

Author : www.technoexam.com [Technowell, Sangli] */

#include <stdio.h>
#include <conio.h>
void main()
{
        int rw, c, no=1 ,len;
        clrscr();
        printf("Enter number of rows: ");
        scanf("%d," &len);
        for(rw=1; rw<=len; rw++)
        {
                printf("\n");
                for(c=1; c<=rw; c++)
                {
                        printf(" %2d ", no);
                        no++;
                }
        }
        getch();
}
```

Output :

```
Enter number of rows: 5

          0
        1 0 1
      2 1 0 1 2
    3 2 1 0 1 2 3
  4 3 2 1 0 1 2 3 4
5 4 3 2 1 0 1 2 3 4 5_
```

Program :

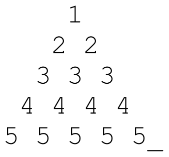
```
/*  Program to print pyramid pattern in C : Pattern 9

Creation Date : 03:19 PM 01/10/2011

Author : www.technoexam.com [Technowell, Sangli] */

#include <stdio.h>
#include <conio.h>
void main()
{
      int no,i,y,x=35;
      clrscr();
      printf("Enter number of rows: ");
      scanf("%d," &no);
      for(y=0;y<=no;y++)
      {
            goto(x,y+1);
            for(i=0-y; i<=y; i++)
            {
                  printf(" %3d ", abs(i));
                  x=x-3;
            }
      }
      getch();
}
```

Output :



```
    1
   2 2
  3 3 3
 4 4 4 4
5 5 5 5 5_
```

Program :

```
/*  Program to print pyramid pattern in C : Pattern 10

Creation Date : 03:14 PM 01/10/2011

Author : www.technoexam.com [Technowell, Sangli] */

#include <stdio.h>
#include <conio.h>
void main()
{
      int i, j=5, k, x;
      clrscr();
      for(i=1;i<=5;i++)
      {
            for(k=1;k<=j;k++)
            {
                  printf(" ");
            }
            for(x=1;x<=i;x++)
```

```
                {
                        printf("%d",i);
                        printf(" ");
                }
                printf("\n");
                j=j-1;
        }
        getch();
}
```

Output :

```
    1
   1 2
  1 2 3
 1 2 3 4
1 2 3 4 5_
```

Program :

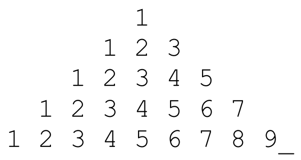
```
/*  Program to print pyramid pattern in C : Pattern 11

Creation Date : 03:24 PM 01/10/2011

Author : www.technoexam.com [Technowell, Sangli] */

#include <stdio.h>
#include <conio.h>
void main()
{
        int rw,c,no,spc;
        clrscr();
        printf("Enter number of rows : ");
        scanf("%d", &no);
        for(rw=1; rw<=no; rw++)
        {
                for(spc=no; spc>=rw; spc--)
                {
                        printf(" ");
                }
                for(c=1; c<=rw; c++)
                {
                        printf("%2d",c);
                }
                printf("\n");
        }
        getch();
}
```

Output :



```
        1
      1 2 3
    1 2 3 4 5
  1 2 3 4 5 6 7
1 2 3 4 5 6 7 8 9_
```

Program :

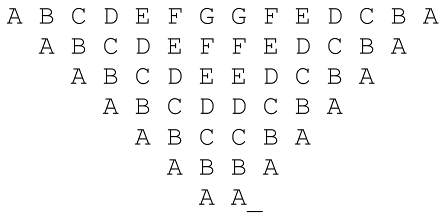
```
/*  Program to print pyramid pattern in C : Pattern 12

Creation Date : 03:24 PM 01/10/2011

Author : www.technoexam.com [Technowell, Sangli] */

#include <stdio.h>
#include <conio.h>
void main()
{
	int i,j,k;
	clrscr();
	for(i=1; i<=5; i++)
	{
		for(j=1; j<=5-i; j++)
		{
			printf("   ");
		}
		for(k=1; k<=2*i-1; k++)
		{
			printf(" %d ",k);
		}
		printf("\n");
	}
	getch();
}
```

Output :



```
 A B C D E F G G F E D C B A
   A B C D E F F E D C B A
     A B C D E E D C B A
       A B C D D C B A
         A B C C B A
           A B B A
             A A_
```

Program :

```
/*  Program to print pyramid pattern in C : Pattern 13

Creation Date : 04:24 PM 01/10/2011

Author : www.technoexam.com [Technowell, Sangli] */

#include <stdio.h>
#include <conio.h>
void main()
```

```
{
       int i,j,asci,spc;
       clrscr();
       for(i=7; i>=1; i--)
       {
              for(spc=6; spc>=i; spc--)
              {
                     printf("  ");
              }
              asci=65;
              for(j=1; j<=i; j++)
              {
                     printf("%2c",asci++);
              }
              for(j=i-1; j>=0; j--)
              {
                     printf("%2c",--asci);
              }
              printf("\n");
       }
       getch();
}
```

Output :

```
AAA AAB AAC ABA ABB ABC ACA ACB ACC BAA BAB BAC BBA BBB
BBC BCA BCB BCC CAA CAB CAC CBA CBB CBC CCA CCB CCC_
```

Program :

```
/* Program to print all Combinations of characters

 A, B, C : Pattern 14

Creation Date : 11:33 PM 01/10/2011

Author : www.technoexam.com [Technowell, Sangli] */

#include <stdio.h>
#include <conio.h>
void main()
{
       char ch1, ch2, ch3;
       clrscr();
       for(ch1='A' ; ch1<='C' ; ++ch1)
       {
              for(ch2='A' ; ch2<='C' ; ++ch2)
              {
                     for(ch3='A' ; ch3<='C' ; ++ch3)
                     {
                     printf(" %c%c%c", ch1, ch2, ch3);
                     }
              }
       }
       getch();
}
```

- Write a C program to print the following pattern:

```
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
```

Program:

```
#include <stdio.h>

int main(void) {
 int i, j;
 for (i = 0; i < 4; i++) {
  for (j = 0; j <= i; j++) {
   if (((i + j) % 2) == 0) {  // Decides on as to which digit to print.
    printf("0");
   } else {
    printf("1");
   }
   printf("\t");
  }
  printf("\n");
 }
 return 0;
}
```

- Write C program to print the following pattern:

```
0
1 1
2 3 5
8 13 21
```

Program:

```
#include <stdio.h>

int main(void) {
 int i, j, a = 0, b = 1, temp = 1;
 for (i = 1; i <= 4; i++) {
  for (j = 1; j <= i; j++) {
   if (i == 1 && j == 1) { // Prints the '0' individually first
    printf("0");
    continue;
   }
   printf("%d ", temp);  // Prints the next digit in the series
   //Computes the series
   temp = a + b;
   a = b;
   b = temp;
   if (i == 4 && j == 3) { // Skips the 4th character of the base
    break;
   }
  }
  printf("\n");
 }
 return 0;
}
```

- Write C program to print the following pattern:

```
1
121
12321
1234321
12321
121
1
```

Program:

```
#include <stdio.h>

void sequence(int x);
int main() {
 /* c taken for columns */
 int i, x = 0, num = 7;
 for (i = 1; i <= num; i++) {
  if (i <= (num / 2) + 1) {
   x = i;
  } else {
   x = 8 - i;
  }
  sequence(x);
  puts("\n");
 }
 return 0;
}

void sequence(int x) {
 int j;

 for (j = 1; j < x; j++) {
  printf("%d", j);
 }
 for (j = x; j > 0; j--) {
  printf("%d", j);
 }
}
```

- Write a C program to print the following pattern:

```
77777777777
         7
        7
       7
      7
     7
    7
   7
  7
 7
7
```

Program:

```
#include <stdio.h>

int main(void) {
 int i, j;
 for (i = 11; i >= 1; i--) {
  for (j = 1; j <= i; j++) {
   if (i == 11) {
    printf("7");  // Makes sure the base is printed completely
    continue;
   } else if (j == i) { // Hollows the rest
    printf("7");
   } else {
    printf(" ");
   }
  }
  printf("\n");
 }
 return 0;
}
```

- Write a C program to print the following pattern:

```
1
2 4
3 6 9
2 4
1
```

Program:

```
#include <stdio.h>
int main(void) {
    int i,j;
    for (i=1; i<=3 ; i++) {
        for (j=1; j<=i; j++)  {
            printf("%2d", (i*j));
        }
        printf("\n");
    }
    for (i=2; i>=1; i--) { // As they share the same base
        for (j=1; j<=i; j++)  {
            printf("%2d",i*j);
        }
        printf("\n");
    }
    return 0;
}
```

- Write a C program to print the following pattern:

```
 1
 1 0
 1 0 0
 1 0 0 0
 1 0 0 0 0
 1 0 0 0 0 0
 1 0 0 0 0 0 0
 1 0 0 0 0 0
 1 0 0 0 0
 1 0 0 0
 1 0 0
 1 0
 1
```

Program:

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

```
int main(void) {
    int i,j;
    for (i=1; i<=7; i++) {
        for (j=1; j<=i; j++) {
            if (j==1) {           // Applying the condition
                printf(" 1");
            } else {
                printf(" 0");
            }
        }
        printf("\n");
    }
    for (i=6; i>=1; i--) {  //As it shares the same base i=6
        for (j=1; j<=i; j++) {
            if (j==1) {   // Applying the condition
                printf(" 1");
            } else {
                printf(" 0");
            }
        }
        printf("\n");
    }
    return 0;
}
```