C PROGRAMS

1.C program to perform all arithmetic operations

```
#include <stdio.h>
int main(void) {
  double num1, num2;
  printf("Enter two numbers: ");
  scanf("%lf %lf", &num1, &num2);
  printf("Sum: %lf\n", num1 + num2);
  printf("Difference: %lf\n", num1 - num2);
  printf("Product: %lf\n", num1 * num2);
  printf("Quotient: %lf\n", num1 / num2);
  return 0;
}
```

```
Enter two numbers: 22
23
Sum: 45.000000
Difference: -1.000000
Product: 506.000000
Quotient: 0.956522
```

2. C program to find area of a triangle if base and height are given

```
#include <stdio.h>
int main(void) {
  double base, height, area;
  printf("Enter the base of the triangle: ");
  scanf("%lf", &base);
  printf("Enter the height of the triangle: ");
  scanf("%lf", &height);
  area = (base * height) / 2;
  printf("The area of the triangle is: %lf\n", area);
  return 0;
```

}

```
Enter the base of the triangle: 5
Enter the height of the triangle: 6
The area of the triangle is: 15.000000
```

3. C program to find all angles of a triangle if two angles are given.

```
#include <stdio.h>
int main(void) {
  double angle1, angle2, angle3;
  printf("Enter the first angle of the triangle: ");
  scanf("%lf", &angle1);
  printf("Enter the second angle of the triangle: ");
  scanf("%lf", &angle2);
  angle3 = 180 - angle1 - angle2;
  printf("The third angle of the triangle is: %lf\n", angle3);
  return 0;
}
```

```
Enter the first angle of the triangle: 60
Enter the second angle of the triangle: 78
The third angle of the triangle is: 42.00000
```

4. C program to convert days in to years, weeks and days.

```
#include <stdio.h>
int main(void) {
  int days, years, weeks, remainingDays;
  printf("Enter the number of days: ");
  scanf("%d", &days);
  years = days / 365;
  weeks = (days % 365) / 7;
  remainingDays = (days % 365) % 7;
  printf("Years: %d\n", years);
  printf("Weeks: %d\n", weeks);
  printf("Days: %d\n", remainingDays);
  return 0;
}
```

```
Enter the number of days: 789
Years: 2
Weeks: 8
Days: 3
```

5. C program to find power and square root of any number.

```
// C program for the above approach
#include <math.h>
#include <stdio.h>

// Function to find the square-root of N
double findSQRT(double N) { return sqrt(N); }

// Driver Code
int main()
{

    // Given number
    int N = 12;

    // Function call
    printf("%f ", findSQRT(N));
    return 0;
}
```

```
Output:

3.464102
```

6. C program to calculate total, average and percentage and grades of five subjects.

```
#include <stdio.h>
int main()
 {
double sub1, sub2, sub3, sub4, sub5; double total, average, percentage;
char grade;
printf("Enter marks of five subjects: ");
```

```
scanf("%lf%lf%lf%lf%lf", &sub1, &sub2, &sub3, &sub4, &sub5);
total = sub1 + sub2 + sub3 + sub4 + sub5;
average = total / 5;
percentage = (total / 500) * 100;
if (percentage >= 90)
   grade = 'A';
else if (percentage >= 80 && percentage < 90)
   grade = 'B';
else if (percentage >= 70 && percentage < 80)
   grade = 'C';
else if (percentage >= 60 && percentage < 70)
   grade = 'D';
else
   grade = 'F';
printf("Total marks: %lf\n", total);
printf("Average marks: %lf\n", average);
printf("Percentage: %lf%%\n", percentage);
printf("Grade: %c\n", grade);
return 0;
}
```

```
Enter marks of five subjects: 90
80
90
99
89
Total marks: 448.000000
Average marks: 89.600000
Percentage: 89.600000%
Grade: B

--------------------------------
Process exited after 10.9 seconds
```

7. C program to check Least Significant Bit (LSB) and MSB of a number using bitwise operator.

```
#include<stdio.h>
Int main ()
{
int num,lsb,msb;
```

```
printf("Enter a number: ");
scanf("%d", &num);
lsb = num & 1;
msb = num >> 31;
printf("Least Significant Bit: %d\n", lsb);
printf("Most Significant Bit: %d\n", msb);
return 0;
}
```

```
Enter a number: 77
Least Significant Bit: 1
Most Significant Bit: 0

--------------------------------
```

8. C program to swap two numbers USING 3RD VARIABLE AND WITHOUT 3RD VARIABLE.

```
#include<stdio.h>
Int main ()
{
 Int num1,num2,temp;
printf("Enter two numbers: ");
scanf("%d%d", &num1, &num2);
temp = num1;
num1 = num2;
num2 = temp;
printf("Swapped numbers using 3rd variable: %d %d\n", num1, num2);
num1 = num1 + num2;
num2 = num1 - num2;
num1 = num1 - num2;
printf("Swapped numbers without 3rd variable: %d %d\n", num1, num2);
return 0;
}
```

```
Enter two numbers: 67
89
Swapped numbers using 3rd variable: 89 67
Swapped numbers without 3rd variable: 67 89
```

9. C program to find maximum between three numbers using

conditional operator AND Ternary Operator.

```
#include<stdio.h>
Int main ()
{
Int num1,num2,num3;
printf("Enter three numbers: ");
scanf("%d%d%d", &num1, &num2, &num3);
int max = (num1 > num2) ? num1 : num2;
max = (max > num3) ? max : num3;
printf("Maximum using conditional operator: %d\n", max);
max = num1 > num2 ? (num1 > num3 ? num1 : num3) : (num2 > num3 ? num2 : num3);
printf("Maximum using ternary operator: %d\n", max);
return 0;
}
```

```
Enter three numbers: 78
90
57
Maximum using conditional operator: 90
Maximum using ternary operator: 90
```

10. C program to check alphabet, digit or special character using Conditional operator.

```
#include<stdio.h>
Int main ()
{
Char ch;
printf("Enter a character: ");
scanf("%c", &ch);
if (ch >= 'A' && ch <= 'Z' || ch >= 'a' && ch <= 'z')
   printf("The character is an alphabet.\n");
else if (ch >= '0' && ch <= '9')
   printf("The character is a digit.\n");
else
   printf("The character is a special character.\n");
return 0;
}
```

```
Enter a character: W
The character is an alphabet.
```

11. C program to calculate total electricity bill

```
#include<stdio.h>
int main()
{
int unit;
double bill;
printf("Enter number of units consumed: ");
scanf("%d", &units);
if (units <= 100)
   bill = units * 0.50;
else if (units <= 200)
   bill = 50 + (units - 100) * 0.75;
else if (units <= 300)
   bill = 125 + (units - 200) * 1.20;
else
   bill = 325 + (units - 300) * 1.50;
printf("Total electricity bill: %.2lf\n", bill);
return 0;
}
```

```
Enter number of units consumed: 77
Total electricity bill: 38.50
```

12. C program to create Simple Calculator AND Days of week

using switch case.

```
#include <stdio.h>

int main() {

  char op;
  double first, second;
  printf("Enter an operator (+, -, *, /): ");
  scanf("%c", &op);
```

```
  printf("Enter two operands: ");
  scanf("%lf %lf", &first, &second);

  switch (op) {
    case '+':
      printf("%.1lf + %.1lf = %.1lf", first, second, first + second);
      break;
    case '-':
      printf("%.1lf - %.1lf = %.1lf", first, second, first - second);
      break;
    case '*':
      printf("%.1lf * %.1lf = %.1lf", first, second, first * second);
      break;
    case '/':
      printf("%.1lf / %.1lf = %.1lf", first, second, first / second);
      break;
    // operator doesn't match any case constant
    default:
      printf("Error! operator is not correct");
  }

  return 0;
}
```

```
Enter an operator (+, -, *,): *
Enter two operands: 1.5
4.5
1.5 * 4.5 = 6.8
```

13. C program to check vowel or consonant using switch case.

```
#include <stdio.h>
int main()
{
    char c;
    printf("Enter a character: ");
```

```
   scanf("%c", &c);
   switch (c)
   {
      case 'a':
      case 'e':
      case 'i':
      case 'o':
      case 'u':
      case 'A':
      case 'E':
      case 'I':
      case 'O':
      case 'U':
         printf("%c is a vowel.\n", c);
         break;
      default:
         printf("%c is a consonant.\n", c);
         break;
   }
   return 0;
}
```

```
Enter a character: w
w is a consonant.
```

14. C program to check positive negative or zero using switch case.

```
#include <stdio.h>
int main()
 {
int num;
printf("Enter a number: ");
scanf("%d", &num);
switch (num > 0)
{
   case 1:
```

```
        printf("The number is positive.\n");
        break;
    case 0:
        switch (num < 0)
        {
            case 1:
                printf("The number is negative.\n");
                break;
            case 0:
                printf("The number is zero.\n");
                break;
        }
        break;
}
return 0;
}
```

```
Enter a number: 78
The number is positive.
```

15. C program to check whether a triangle is Equilateral, Isosceles or Scalene.

```
include <stdio.h>
int main()
{
 int side1, side2, side3
printf("Enter three sides of a triangle: ");
scanf("%d%d%d", &side1, &side2, &side3);
if (side1 == side2 && side2 == side3)
   printf("The triangle is Equilateral.\n");
else if (side1 == side2 || side2 == side3 || side1 == side3)
   printf("The triangle is Isosceles.\n");
else
   printf("The triangle is Scalene.\n");
return 0;
}
```

```
Enter three sides of a triangle: 89
90
67
The triangle is Scalene.
```

16. C program to print all natural numbers AND sum of it from 1

to n.

```
#include <stdio.h>

int main()

{

int n, i, sum = 0;

printf("Enter a number: ");

scanf("%d", &n);

printf("All natural numbers from 1 to %d: ", n);

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

{

   printf("%d ", i);

   sum += i;

}

printf("\n");

printf("Sum of all natural numbers from 1 to %d: %d\n", n, sum);

return 0;

}
```

```
Enter a number: 56
All natural numbers from 1 to 56: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56
Sum of all natural numbers from 1 to 56: 1596
```

17. C program to print all even numbers AND sum of it from 1 to n

```
#include <stdio.h>

int main()

 {

int n, i, sum = 0;

printf("Enter a number: ");

scanf("%d", &n);

printf("All even numbers from 1 to %d: ", n);

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

{

   printf("%d ", i);
```

```
    sum += i;
}
printf("\n");

printf("Sum of all even numbers from 1 to %d: %d\n", n, sum);
return 0;
}
```

```
Enter a number: 67
All even numbers from 1 to 67: 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62
64 66
Sum of all even numbers from 1 to 67: 1122
```

18. C program to print multiplication table of a number.

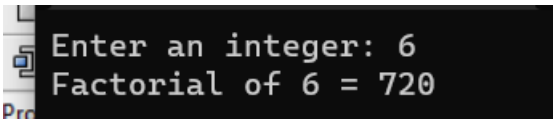
```
#include <stdio.h>
int main()
{
    int n, i;
    printf("Enter a number: ");
    scanf("%d", &n);
    for (i = 1; i <= 10; i++)
    {
        printf("%d * %d = %d\n", n, i, n * i);
    }
    return 0;
}
```

```
Enter a number: 67
67 * 1 = 67
67 * 2 = 134
67 * 3 = 201
67 * 4 = 268
67 * 5 = 335
67 * 6 = 402
67 * 7 = 469
67 * 8 = 536
67 * 9 = 603
67 * 10 = 670
```

19. C program to calculate factorial of a number**.**

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

```
{
  int n, i;
  long long factorial = 1;
  printf("Enter an integer: ");
  scanf("%d", &n);
  if (n < 0)
    printf("Error! Factorial of a negative number doesn't exist.");
#  else
  {
    for (i = 1; i <= n; ++i)
    {
      factorial *= i;
    }
    printf("Factorial of %d = %lld", n, factorial);
  }
  return 0;
}
```



20. C program to check whether a number is palindrome or not.

```
#include <stdio.h>
int main()
{
  int n, reversedN = 0, remainder, originalN;
  printf("Enter an integer: ");
  scanf("%d", &n);
  originalN = n;
  // reversed integer is stored in reversedN
  while (n != 0)
  {
    remainder = n % 10;
    reversedN = reversedN * 10 + remainder;
    n /= 10;
  }
```

```
 // palindrome if originalN and reversedN are equal
 if (originalN == reversedN)
  printf("%d is a palindrome.", originalN);
 else
  printf("%d is not a palindrome.", originalN);
 return 0;
}
```

```
Enter an integer: 6
6 is a palindrome.
```

21. C program to count frequency of digits in a given number.

```
Enter an integer: 6
6 is a palindrome.
```

22. C program to find HCF(GCD) AND LCM of two numbers.

```
#include <stdio.h>
int main()
{
 int n1, n2;
 int hcf, lcm;
 printf("Enter two positive integers: ");
 scanf("%d %d", &n1, &n2);
 // maximum number between n1 and n2 is stored in hcf
 hcf = (n1 > n2) ? n1 : n2;
 while (1)
 {
  if (hcf % n1 == 0 && hcf % n2 == 0)
  {
   printf("The LCM of %d and %d is %d.\n", n1, n2, hcf);
   break;
  }
  ++hcf;
 }
 // minimum number between n1 and n2 is stored in lcm
 lcm = (n1 < n2) ? n1 : n2;
 while (1)
```

```
 {
  if (lcm % n1 == 0 && lcm % n2 == 0)
  {
    printf("The HCF of %d and %d is %d.", n1, n2, lcm);
    break;
  }
  ++lcm;
 }
 return 0;
}
```

```
Enter two positive integers: 2
8
The LCM of 2 and 8 is 8.
The HCF of 2 and 8 is 8.
```

23. C program to print all prime numbers between 1 to n.

```
#include <stdio.h>
int main()
{
 int n, i, flag;
 printf("Enter a positive integer: ");
 scanf("%d", &n);
 printf("Prime numbers between 1 and %d are: ", n);
 for (i = 2; i <= n; ++i)
 {
  flag = 1;
  for (int j = 2; j <= i / 2; ++j)
  {
    if (i % j == 0)
    {
      flag = 0;
      break;
    }
  }
  if (flag == 1)
    printf("%d ", i);
```

```
 }
 return 0;
}
```

```
Enter a positive integer: 45
Prime numbers between 1 and 45 are: 2 3 5 7 11 13 17 19 23 29 31 37 41 43
```

24. C program to print all Strong Numbers between 1 to n

```
#include <stdio.h>
int main()
{
 int n, i, j, sum, fact, originalN;
 printf("Enter a positive integer: ");
 scanf("%d", &n);
 printf("Strong numbers between 1 and %d are: ", n);
 for (i = 1; i <= n; ++i)
 {
  originalN = i;
  sum = 0;
  while (i > 0)
  {
   fact = 1;
   int lastDigit = i % 10;
   for (j = 1; j <= lastDigit; ++j)
    fact *= j;
   sum += fact;
   i /= 10;
  }
  if (sum == originalN)
   printf("%d ", originalN);
  i = originalN;
 }
 return 0;
}
```

```
Enter a positive integer: 55
Strong numbers between 1 and 55 are: 1 2
```

25. C program to print Fibonacci series up to n terms.

```
#include <stdio.h>
int main()
{
 int i, n, t1 = 0, t2 = 1, nextTerm;
 printf("Enter the number of terms: ");
 scanf("%d", &n);
 printf("Fibonacci Series: ");
 for (i = 1; i <= n; ++i)
 {
  printf("%d, ", t1);
  nextTerm = t1 + t2;
  t1 = t2;
  t2 = nextTerm;
 }
 return 0;
}
```

```
Enter the number of terms: 6
Fibonacci Series: 0, 1, 1, 2, 3, 5,
```

26. C program to print Armstrong numbers from 1 to n AND

Check a given number is Armstrong numbers or not.

```
#include <stdio.h>
int main()
{
 int n, i, originalN, remainder, result;
 // Print Armstrong numbers from 1 to n
 printf("Enter the upper limit: ");
 scanf("%d", &n);
 printf("Armstrong numbers between 1 and %d are: ", n);
 for (i = 1; i <= n; ++i)
 {
  originalN = i;
  result = 0;
  while (originalN != 0)
  {
```

```
    remainder = originalN % 10;
    result += remainder * remainder * remainder;
    originalN /= 10;
   }
   if (result == i)
    printf("%d ", i);
 }
 // Check if a given number is an Armstrong number or not
 int num;
 printf("\nEnter a number to check if it is an Armstrong number: ");
 scanf("%d", &num);
 originalN = num;
 result = 0;
 while (originalN != 0)
 {
  remainder = originalN % 10;
  result += remainder * remainder * remainder;
  originalN /= 10;
 }
 if (result == num)
  printf("%d is an Armstrong number.", num);
 else
  printf("%d is not an Armstrong number.", num);
 return 0;
}
```

```
Enter the upper limit: 6
Armstrong numbers between 1 and 6 are: 1
Enter a number to check if it is an Armstrong number: 7
7 is not an Armstrong number.
```

27. C program to print all Perfect numbers between 1 to n AND

Check a given number is Perfect numbers or not.

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

```
  int n, i, sum;
  // Print all Perfect numbers between 1 to n
  printf("Enter the upper limit: ");
  scanf("%d", &n);
  printf("Perfect numbers between 1 and %d are: ", n);
  for (i = 1; i <= n; ++i)
  {
    sum = 0;
    for (int j = 1; j < i; ++j)
    {
      if (i % j == 0)
        sum += j;
    }
    if (sum == i)
      printf("%d ", i);
  }
  // Check if a given number is a Perfect number or not
  int num;
  printf("\nEnter a number to check if it is a Perfect number: ");
  scanf("%d", &num);
  sum = 0;
  for (i = 1; i < num; ++i)
  {
    if (num % i == 0)
      sum += i;
  }
  if (sum == num)
    printf("%d is a Perfect number.", num);
  else
    printf("%d is not a Perfect number.", num);
  return 0;
}
```

```
Enter the upper limit: 6
Perfect numbers between 1 and 6 are: 6
Enter a number to check if it is a Perfect number: 78
78 is not a Perfect number.
```

28. C program to find power of any number using for loop.

```
#include <stdio.h>

int main()

{

 int base, exponent;

 long long result = 1;

 printf("Enter base and exponent: ");

 scanf("%d %d", &base, &exponent);


 for (int i = 1; i <= exponent; ++i)

 {

  result *= base;

 }

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

 return 0;

}
```

```
Enter base and exponent: 4
3
4 ^ 3 = 64
```

29. C program to print ASCII values of all characters.

```
#include <stdio.h>

int main()

{

 char c;

 printf("Enter a character: ");

 scanf("%c", &c);

 printf("ASCII value of %c = %d", c, c);

 return 0;

}
```

```
Enter a character: e
ASCII value of e = 101
------------------------------
```

30. C program to print Pascal triangle up to n rows.

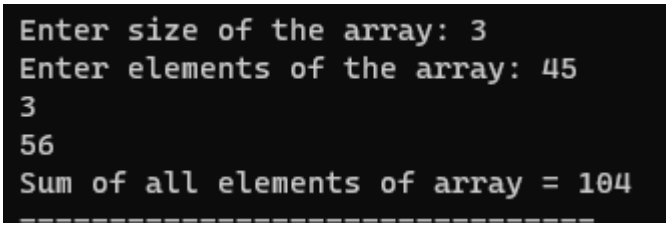
```
#include <stdio.h>

int main()

{

 int rows, coef = 1, space, i, j;

 printf("Enter number of rows: ");

 scanf("%d", &rows);

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

 {

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

   printf("  ");

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

  {

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

    coef = 1;

   else

    coef = coef * (i - j + 1) / j;

   printf("%4d", coef);

  }

  printf("\n");

 }

 return 0;

}
```

```
Enter number of rows: 3
          1
        1   1
      1   2   1
```

31. C program to find sum of all elements of array.

```
#include <stdio.h>

int main()

{

 int size, i, sum = 0;
```

```
 int arr[100];
 printf("Enter size of the array: ");
 scanf("%d", &size);
 printf("Enter elements of the array: ");
 for (i = 0; i < size; ++i)
 {
  scanf("%d", &arr[i]);
  sum += arr[i];
 }
 printf("Sum of all elements of array = %d", sum);
 return 0;
}
```



32. C program to copy one array to another array.

```
#include <stdio.h>
int main()
{
 int size, i;
 int source[100], target[100];
 printf("Enter size of the array: ");
 scanf("%d", &size);
 printf("Enter elements of the source array: ");
 for (i = 0; i < size; ++i)
 {
  scanf("%d", &source[i]);
 }
 // Copying elements of source array to target array
 for (i = 0; i < size; ++i)
 {
  target[i] = source[i];
```

```
  }
  printf("Elements of target array: ");
  for (i = 0; i < size; ++i)
  {
    printf("%d ", target[i]);
  }
  return 0;
}
```

```
Enter size of the array: 3
Enter elements of the source array: 3
4
3
Elements of target array: 3 4 3
--------------------------------
```

33. C program to insert an element in array at specified position.

```
#include <stdio.h>
int main()
{
  int size, i, pos, element;
  int arr[100];
  printf("Enter size of the array: ");
  scanf("%d", &size);
  printf("Enter elements of the array: ");
  for (i = 0; i < size; ++i)
  {
    scanf("%d", &arr[i]);
  }
  printf("Enter the element to be inserted: ");
  scanf("%d", &element);
  printf("Enter the position where the element is to be inserted: ");
  scanf("%d", &pos);

  // Shift elements of arr[pos...size-1] right by 1
  for (i = size - 1; i >= pos; --i)
  {
    arr[i + 1] = arr[i];
```

```
 }
 // Insert element at given position
 arr[pos] = element;
 printf("Array after inserting element at given position:\n");
 for (i = 0; i < size + 1; ++i)
 {
  printf("%d ", arr[i]);
 }
 return 0;
}
```

```
Enter size of the array: 3
Enter elements of the array: 23
45
67
Enter the element to be inserted: 34
Enter the position where the element is to be inserted: 3
Array after inserting element at given position:
23 45 67 34
```

34. C program to delete an element in array at specified position.

```
#include <stdio.h>
int main()
{
 int size, i, pos;
 int arr[100];
 printf("Enter size of the array: ");
 scanf("%d", &size);
 printf("Enter elements of the array: ");
 for (i = 0; i < size; ++i)
 {
  scanf("%d", &arr[i]);
 }
 printf("Enter the position where the element is to be deleted: ");
 scanf("%d", &pos);
 // Shift elements of arr[pos+1...size-1] left by 1
 for (i = pos; i < size - 1; ++i)
 {
```

```
    arr[i] = arr[i + 1];
  }
  printf("Array after deleting element at given position:\n");
  for (i = 0; i < size - 1; ++i)
  {
    printf("%d ", arr[i]);
  }
  return 0;
}
```

```
Enter size of the array: 3
Enter elements of the array: 45
65
76
Enter the position where the element is to be deleted: 2
Array after deleting element at given position:
45 65
--------------------------------
```

35. C program to search element in array using Linear Search.

```
#include <stdio.h>
int main()
{
  int size, i, element, found = 0;
  int arr[100];
  printf("Enter size of the array: ");
  scanf("%d", &size);
  printf("Enter elements of the array: ");
  for (i = 0; i < size; ++i)
  {
    scanf("%d", &arr[i]);
  }
  printf("Enter the element to be searched: ");
  scanf("%d", &element);
  for (i = 0; i < size; ++i)
  {
    if (arr[i] == element)
    {
      found = 1;
```

```
    printf("Element %d found at position %d\n", element, i + 1);
    break;
   }
  }
 if (found == 0)
  printf("Element %d not found in the array\n", element);
 return 0;
}
```

```
Enter size of the array: 3
Enter elements of the array: 23
34
28
Enter the element to be searched: 23
Element 23 found at position 1
```

36. C program to find second largest number and Sorting Using Bubble sort in an array.

```
#include <stdio.h>
int main()
{
 int size, i, j, temp;
 int arr[100];
 printf("Enter size of the array: ");
 scanf("%d", &size);
 printf("Enter elements of the array: ");
 for (i = 0; i < size; ++i)
 {
  scanf("%d", &arr[i]);
 }
 // Find second largest number
 int max1 = 0, max2 = 0;
 for (i = 0; i < size; ++i)
 {
  if (arr[i] > max1)
  {
   max2 = max1;
```

```
    max1 = arr[i];
   }
   else if (arr[i] > max2 && arr[i] < max1)
   {
    max2 = arr[i];
   }
  }
  printf("Second largest number = %d\n", max2);
  // Sort the array using Bubble sort
  for (i = 0; i < size - 1; ++i)
  {
   for (j = 0; j < size - i - 1; ++j)
   {
    if (arr[j] > arr[j + 1])
    {
     temp = arr[j];
     arr[j] = arr[j + 1];
     arr[j + 1] = temp;
    }
   }
  }
  printf("Array after sorting:\n");
  for (i = 0; i < size; ++i)
  {
   printf("%d ", arr[i]);
  }
  return 0;
}
```

```
Enter size of the array: 4
Enter elements of the array: 23
2
3
4
Second largest number = 4
Array after sorting:
2 3 4 23
```

37. C program to count total number of duplicate elements in an array.

```
#include <stdio.h>
int main()
{
  int size, i, j, count = 0;
  int arr[100];
  printf("Enter size of the array: ");
  scanf("%d", &size);
  printf("Enter elements of the array: ");
  for (i = 0; i < size; ++i)
  {
    scanf("%d", &arr[i]);
  }
  for (i = 0; i < size; ++i)
  {
    for (j = i + 1; j < size; ++j)
    {
      if (arr[i] == arr[j])
      {
        ++count;
        break;
      }
    }
  }
  printf("Total number of duplicate elements in the array = %d", count);
  return 0;
}
```

```
Enter size of the array: 3
Enter elements of the array: 23
23
34
Total number of duplicate elements in the array = 1
--------------------------------
```

38. 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;
}
```

```
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. C program to find sum of main diagonal elements of a matrix.

```
#include <stdio.h>
int main()
{
 int rows, cols, i, j;
 int matrix[100][100];
 int sum = 0;
 printf("Enter number of rows: ");
 scanf("%d", &rows);
 printf("Enter number of columns: ");
 scanf("%d", &cols);
 printf("Enter elements of the matrix:\n");
 for (i = 0; i < rows; ++i)
 {
  for (j = 0; j < cols; ++j)
  {
```

```
    scanf("%d", &matrix[i][j]);
   }
  }
  for (i = 0; i < rows; ++i)
  {
   for (j = 0; j < cols; ++j)
   {
    if (i == j)
     sum += matrix[i][j];
   }
  }
  printf("Sum of main diagonal elements = %d", sum);
  return 0;
}
```

```
Enter number of rows: 2
Enter number of columns: 3
Enter elements of the matrix:
2
3
4
23
34
43
Sum of main diagonal elements = 36
```

40. C program to check sparse AND transpose matrix.

```
#include <stdio.h>
int main()
{
  int rows, cols, i, j;
  int matrix[100][100];
  printf("Enter number of rows: ");
  scanf("%d", &rows);
  printf("Enter number of columns: ");
  scanf("%d", &cols);
  printf("Enter elements of the matrix:\n");
  for (i = 0; i < rows; ++i)
  {
```

```
  for (j = 0; j < cols; ++j)
  {
    scanf("%d", &matrix[i][j]);
  }
 }
 // Check if matrix is sparse
 int count = 0;
 for (i = 0; i < rows; ++i)
 {
  for (j = 0; j < cols; ++j)
  {
    if (matrix[i][j] == 0)
      ++count;
  }
 }
 if (count > (rows * cols) / 2)
  printf("Matrix is sparse\n");
 else
  printf("Matrix is not sparse\n");
 // Find transpose of matrix
 printf("Transpose of the matrix:\n");
 for (i = 0; i < rows; ++i)
 {
  for (j = 0; j < cols; ++j)
  {
    printf("%d ", matrix[j][i]);
  }
  printf("\n");
 }
 return 0;
}
```

```
Enter number of rows: 4
Enter number of columns: 3
Enter elements of the matrix:
23
34
22
32
23
34
32
43
23
23
2
3
Matrix is not sparse
Transpose of the matrix:
23 32 32
34 23 43
22 34 23
0 0 0
```

41. C program to check whether a matrix is Identity matrix or not.

#include<stdio.h>

```
int main()
{
   int i, j, rows, columns, a[10][10], Flag = 1;

   printf("\n Please Enter Number of rows and columns  :  ");
   scanf("%d %d", &i, &j);

   printf("\n Please Enter the Matrix Elements \n");
   for(rows = 0; rows < i; rows++)
   {
      for(columns = 0; columns < j; columns++)
      {
         scanf("%d", &a[rows][columns]);
      }
   }

   for(rows = 0; rows < i; rows++)
   {
      for(columns = 0; columns < j; columns++)
```

```
    {
      if(a[rows][columns] != 1 && a[columns][rows] != 0)
      {
        Flag = 0;
        break;
      }
    }
  }
  if(Flag == 1)
  {
    printf("\n The Matrix that you entered is an Identity Matrix ");
  }
  else
  {
    printf("\n The Matrix that you entered is Not an Identity Matrix ");
  }
  return 0;
}
```

```
 Please Enter Number of rows and columns  :  3
4

 Please Enter the Matrix Elements
23
45
67
76
54
67
67
78
76
65
4
3

 The Matrix that you entered is Not an Identity Matrix
```

42. C program to merge two sorted array in ascending order.

```
#include <stdio.h>



int main() {
 //Declaring the size of arrays
 int s1, s2, s3;
 printf("\n Enter the size of 1st array  ");
```

```
scanf("%d", & s1);
printf("\n Enter the size of 2nd array ");
scanf("%d", & s2);



s3 = s1 + s2;
printf("\n Enter the elements of 1st array\n");
// Declaring the array
int arr1[s1], arr2[s2], arr3[s3];
//Initialising the array
for (int i = 0; i < s1; i++) {
  scanf("%d", & arr1[i]);
  arr3[i] = arr1[i];
}
int k = s1;
printf("\nEnter the elements of 2nd array \n");
for (int i = 0; i < s2; i++) //Array Initialised
{
  scanf("%d", & arr2[i]);
  arr3[k] = arr3[i];
  k++;
}
printf("\nThe merged array before sorting : \n\t");
for (int i = 0; i < s3; i++)
  printf("%d ", arr3[i]); //Print the merged array before sorting



printf("\n The merged array after sorting\n\t");
for (int i = 0; i < s3; i++) //Sorting the array
{
  int tem;
  for (int j = i + 1; j < s3; j++) {
    if (arr3[i] > arr3[j]) {
      tem = arr3[i];
```

```
      arr3[i] = arr3[j];
      arr3[j] = tem;
     }
    }
  }

  for (int i = 0; i < s3; i++) //Printing the sorted Array
  {
   printf(" %d ", arr3[i]);
  }
}
```

```
 Enter the size of 1st array  3

 Enter the size of 2nd array 4

 Enter the elements of 1st array
12
34
45

Enter the elements of 2nd array
23
45
67
87

The merged array before sorting :
        12 34 45 12 34 45 12
 The merged array after sorting
         12  12  12  34  34  45  45
```

#

43. All Operations of String.

1. **strlen**: This function returns the length of a string.
2. **strcpy**: This function copies one string to another.
3. **strcat**: This function concatenates two strings.
4. **strcmp**: This function compares two strings and returns an integer based on the result of the comparison.
5. **strchr**: This function searches for a particular character in a string and returns a pointer to the first occurrence of the character.
6. **strstr**: This function searches for a particular substring within a string and returns a pointer to the first occurrence of the substring.
7. **strtok**: This function breaks a string into tokens based on a specified delimiter.

8. **strspn**: This function returns the length of the initial portion of a string that consists only of characters from a specified set.
9. **strpbrk**: This function searches a string for any of a set of characters and returns a pointer to the first occurrence of any of those characters.
10. **strcoll**: This function compares two strings using the current locale's collating sequence.

44. C program to check whether a string is palindrome or not

without Compare Function of String.

```
#include <stdio.h>
#include <string.h>
int main()
{
   char str[100];
   int i, length;
   int flag = 0;
   printf("Enter a string: ");
   scanf("%s", str);
   length = strlen(str);
   for(i=0; i < length ;i++)
   {
      if(str[i] != str[length-i-1])
      {
         flag = 1;
         break;
      }
   }
   if (flag)
   {
      printf("%s is not a palindrome", str);
   }
   else
   {
      printf("%s is a palindrome", str);
   }
   return 0;
```

}

```
Enter a string: r
r is a palindrome
```

45. C program to count frequency of each character in a string.

```
#include <stdio.h>
#include <string.h>
int main()
{
   char str[100];
   int count[256] = {0};
   int i;
   printf("Enter a string: ");
   scanf("%[^\n]s", str);
   for (i = 0; str[i] != '\0'; i++)
   {
      count[str[i]]++;
   }
   for (i = 0; i < 256; i++)
   {
      if (count[i] > 0)
      {
         printf("'%c' occurs %d times\n", i, count[i]);
      }
   }
   return 0;
}
```

```
Enter a string: a
'a' occurs 1 times
```

46. C program to find diameter, circumference and area of a circle

using functions.

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

```
#include <math.h>
#define PI 3.14159
// Function prototypes
float diameter(float radius);
float circumference(float radius);
float area(float radius);
int main()
{
   float radius, d, c, a;
   printf("Enter the radius of the circle: ");
   scanf("%f", &radius);
   d = diameter(radius);
   c = circumference(radius);
   a = area(radius);
   printf("Diameter: %.2f\n", d);
   printf("Circumference: %.2f\n", c);
   printf("Area: %.2f\n", a);
   return 0;
}
float diameter(float radius)
{
   return 2 * radius;
}
float circumference(float radius)
{
   return 2 * PI * radius;
}
float area(float radius)
{
   return PI * radius * radius;
}
```

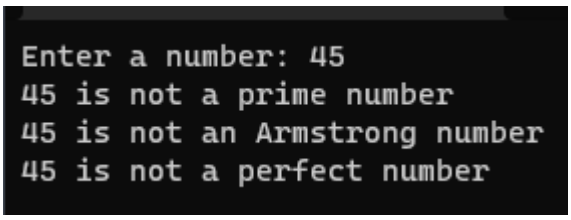
```
Enter the radius of the circle: 81
Diameter: 162.00
Circumference: 508.94
Area: 20611.97
```

47. C program to check prime, armstrong and perfect numbers using functions.

```
#include <stdio.h>
#include <math.h>
int is_prime(int n);
int is_armstrong(int n);
int is_perfect(int n);
int main()
{
   int n, prime, armstrong, perfect;
   printf("Enter a number: ");
   scanf("%d", &n);
   prime = is_prime(n);
   armstrong = is_armstrong(n);
   perfect = is_perfect(n);
   if (prime)
   {
      printf("%d is a prime number\n", n);
   }
   else
   {
      printf("%d is not a prime number\n", n);
   }
   if (armstrong)
   {
      printf("%d is an Armstrong number\n", n);
   }
   else
   {
      printf("%d is not an Armstrong number\n", n);
```

```
    }
    if (perfect)
    {
        printf("%d is a perfect number\n", n);
    }
    else
    {
        printf("%d is not a perfect number\n", n);
    }
    return 0;
}
int is_prime(int n)
{
    int i;
    if (n <= 1)
    {
        return 0;
    }
    for (i = 2; i <= sqrt(n); i++)
    {
        if (n % i == 0)
        {
            return 0;
        }
    }
    return 1;
}
int is_armstrong(int n)
{
    int original, rem, result = 0, digits = 0;
    original = n;
    // Count the number of digits
    while (original != 0)
    {
```

```
        original /= 10;
        digits++;
    }
    original = n;
    // Check if the number is an Armstrong number
    while (original != 0)
    {
        rem = original % 10;
        result += pow(rem, digits);
        original /= 10;
    }
    return (n == result);
}
int is_perfect(int n)
{
    int i, sum = 0;
    for (i = 1; i < n; i++)
    {
        if (n % i == 0)
        {
            sum += i;
        }
    }
    return (n == sum);
}
```



48. C program to add two number using pointers.

```
#include <stdio.h>
int main()
{
    int a, b, sum;
```

```
    int *pa, *pb;
    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);
    pa = &a;
    pb = &b;

    sum = *pa + *pb;
    printf("Sum = %d\n", sum);
    return 0;
}
```

```
Enter two numbers: 56
78
Sum = 134
```

49. Swap 2 numbers using Call by Value AND Call by reference.

```
#include <stdio.h>
void swap_by_value(int x, int y);
void swap_by_reference(int *x, int *y);
int main()
{
    int a, b;
    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);
    printf("Before swapping (call by value): a = %d, b = %d\n", a, b);
    swap_by_value(a, b);
    printf("After swapping (call by value): a = %d, b = %d\n", a, b);
    printf("Before swapping (call by reference): a = %d, b = %d\n", a, b);
    swap_by_reference(&a, &b);
    printf("After swapping (call by reference): a = %d, b = %d\n", a, b);
    return 0;
}
void swap_by_value(int x, int y)
{
    int temp;
```

```
    temp = x;
    x = y;
    y = temp;
}
void swap_by_reference(int *x, int *y)
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}
```

```
Enter two numbers: 90
89
Before swapping (call by value): a = 90, b = 89
After swapping (call by value): a = 90, b = 89
Before swapping (call by reference): a = 90, b = 89
After swapping (call by reference): a = 89, b = 90
```

50. C program to copy an array to another array AND reverse an

array using pointers.

```
#include <stdio.h>

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;
    printf("Enter size of array: ");
    scanf("%d", &size);
    printf("Enter elements in array: ");
    for (i = 0; i < size; i++)
    {
        scanf("%d", (source_ptr + i));
    }
    end_ptr = &source_arr[size - 1];
    printf("\nSource array before copying: ");
    printArray(source_arr, size);

    printf("\nDestination array before copying: ");
    printArray(dest_arr, size);
    while(source_ptr <= end_ptr)
    {
        *dest_ptr = *source_ptr;
```

```
        // Increment source_ptr and dest_ptr
        source_ptr++;
        dest_ptr++;
    }
    printf("\n\nSource array after copying: ");
    printArray(source_arr, size);

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

    return 0;
}
void printArray(int *arr, int size)
{
    int i;

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

PATTERNS

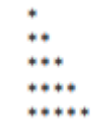
1. NUMBER PATTERN 1

```
11111
11111
11111
11111
11111
```

```
#include <stdio.h>

int main()

{

    int i, j, N;

    printf("Enter number of rows: ");

    scanf("%d", &N);

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

    {

        for(j=1; j<=N; j++)

        {

            printf("*");

        }

        printf("\n");
```

```
    }
    return 0;
}
```

2 RIGHT TRIANGLE



```
#include <stdio.h>
int main()
{
    int i, j, n;
    printf("Enter value of n: ");
    scanf("%d", &n);
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=i; j++)
        {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

3 MIRRORED RIGHT

```
#include <stdio.h>
int main()
{
    int i, j, rows;
    printf("Enter number of rows: ");
    scanf("%d", &rows);
    for(i=1; i<=rows; i++)
```

```
    {
        for(j=i; j<rows; j++)
        {
            printf(" ");
        }
        for(j=1; j<=i; j++)
        {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

4 PYRAMIDAL

```
#include <stdio.h>
int main()
{
    int i, j, rows;
    printf("Enter number of rows : ");
    scanf("%d", &rows);
    for(i=1; i<=rows; i++)
    {
        for(j=i; j<rows; j++)
        {
            printf(" ");
        }
        for(j=1; j<=(2*i-1); j++)
        {
            printf("*");
        }
        printf("\n");
```

```
    }
    return 0;
}
```

5 SQUARE 1

```
#include <stdio.h>
int main()
{
    int rows, cols, i, j;
    printf("Enter number of rows: ");
    scanf("%d", &rows);
    printf("Enter number of columns: ");
    scanf("%d", &cols);
    for(i=1; i<=rows; i++)
    {
        for(j=1; j<=cols; j++)
        {
            printf("1");
        }
        printf("\n");
    }
    return 0;
}
```