

### **1. C program to perform all arithmetic operations**

```
#include <stdio.h>

int main(void) {
    double num1, num2;
    char op;
    printf("Enter an operator (+, -, *, /): ");
    scanf("%c", &op);
    printf("Enter two numbers: ");
    scanf("%lf %lf", &num1, &num2);
    if (op == '+') {
        printf("%.1lf + %.1lf = %.1lf\n", num1, num2, num1 + num2);
    } else if (op == '-') {
        printf("%.1lf - %.1lf = %.1lf\n", num1, num2, num1 - num2);
    } else if (op == '*') {
        printf("%.1lf * %.1lf = %.1lf\n", num1, num2, num1 * num2);
    } else if (op == '/') {
        printf("%.1lf / %.1lf = %.1lf\n", num1, num2, num1 / num2);
    } else {
        printf("Error: Invalid operator\n");
    }
    return 0;
}
```

### **OUTPUT**

Enter an operator (+, -, \*, /): +

Enter two numbers: 10

20

10.0 + 20.0 = 30.0

### **2. 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 = %.2f sq. units", area);
    return 0;
}

```

**OUTPUT**

```

Enter base of the triangle: 20
Enter height of the triangle: 15
Area of the triangle = 150.00 sq. units

```

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

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

```

int main(void) {
    int angle1, angle2, angle3;

    printf("Enter the first angle of the triangle: ");
    scanf("%d", &angle1);

    printf("Enter the second angle of the triangle: ");
    scanf("%d", &angle2);

    angle3 = 180 - (angle1 + angle2);
    printf("The third angle of the triangle is: %d\n", angle3);

    return 0;
}

```

**OUTPUT**

```

Enter the first angle of the triangle: 70
Enter the second angle of the triangle: 65
The third angle of the triangle is: 45

```

**4. C program to convert days into years, weeks and days.**

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

```

int main()
{

```

```

int days, years, weeks;

printf("Enter the number of days: ");
scanf("%d", &days);

years = days / 365;
weeks = (days % 365) / 7;
days = (days % 365) % 7;
printf("Years: %d\n", years);
printf("Weeks: %d\n", weeks);
printf("Days: %d\n", days);
return 0;
}

```

**OUTPUT**

```

Enter the number of days: 816
Years: 2
Weeks: 12
Days: 2

```

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

```

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

int main()
{
    double num, power, square_root;

    printf("Enter a number: ");
    scanf("%lf", &num);

    power = pow(num, 2);
    square_root = sqrt(num);

    printf("Square: %.2lf\n", power);
    printf("Square root: %.2lf\n", square_root);

    return 0;
}

```

**OUTPUT**

```

Enter a number: 49
Square: 2401.00
Square root: 7.00

```

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

```
#include <stdio.h>
int main() {
    float marks[5];
    float total = 0;
    float average;
    float percentage;
    int i;
    for (i = 0; i < 5; i++) {
        printf("Enter marks for subject %d: ", i+1);
        scanf("%f", &marks[i]);
        total += marks[i];
    }
    average = total / 5;
    percentage = (total / 500) * 100;
    printf("Total marks: %.2f\n", total);
    printf("Average marks: %.2f\n", average);
    printf("Percentage: %.2f\n", percentage);
    if (percentage >= 90) {
        printf("Grade: A+\n");
    } else if (percentage >= 80) {
        printf("Grade: A\n");
    } else if (percentage >= 70) {
        printf("Grade: B+\n");
    } else if (percentage >= 60) {
        printf("Grade: B\n");
    } else if (percentage >= 50) {
        printf("Grade: C+\n");
    } else if (percentage >= 40) {
        printf("Grade: C\n");
    } else {
        printf("Grade: D\n");
    }
    return 0;
}
```

#### **OUTPUT**

```
Enter marks for subject 1: 90
Enter marks for subject 2: 85
Enter marks for subject 3: 95
Enter marks for subject 4: 80
Enter marks for subject 5: 78
Total marks: 428.00
Average marks: 85.60
Percentage: 85.60
```

Grade: A

**7. C program to check least significant Bit (LSB) and MSB of a number using bitwise operator.**

```
#include <stdio.h>
int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    if (num & 1) {
        printf("LSB is set (1).\n");
    } else {
        printf("LSB is not set (0).\n");
    }
    if (num & (1 << (sizeof(int) * 8 - 1))) {
        printf("MSB is set (1).\n");
    } else {
        printf("MSB is not set (0).\n");
    }
    return 0;
}
```

**OUTPUT**

Enter a number: 1

LSB is set (1).

MSB is not set (0).

**8. C program to swap two number using 3<sup>rd</sup> variable and without 3<sup>rd</sup> variable.**

```
#include <stdio.h>
int main() {
    int a, b, temp;
    printf("Enter two numbers: ");
    scanf("%d%d", &a, &b);
    printf("Before swapping: a = %d, b = %d\n", a, b);
    temp = a;
    a = b;
    b = temp;
    printf("After swapping: a = %d, b = %d\n", a, b);
    return 0;
}
```

**OUTPUT**

Enter two numbers: 5

10

Before swapping: a = 5, b = 10

After swapping: a = 10, b = 5

**9. C program to find maximum between three number using conditional operator and ternary operator.**

```
#include <stdio.h>
int main() {
    int a, b, c;
    printf("Enter three numbers: ");
    scanf("%d%d%d", &a, &b, &c);
    int max = a > b ? (a > c ? a : c) : (b > c ? b : c);
    printf("Maximum: %d\n", max);
    return 0;
}
```

**OUTPUT**

```
Enter three numbers: 20
```

```
50
```

```
10
```

```
Maximum: 50
```

**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);
    (ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z') ? printf("\'%c\' is an alphabet.\n", ch) :
    (ch >= '0' && ch <= '9') ? printf("\'%c\' is a digit.\n", ch) :
    printf("\'%c\' is a special character.\n", ch);
    return 0;
}
```

**OUTPUT**

```
Enter a character: A
```

```
'A' is an alphabet.
```

**11. C program to calculate total electricity bill.**

```
#include <stdio.h>
int main()
{
    int units;
    float bill;
    printf("Enter the number of units consumed: ");
    scanf("%d", &units);
    if (units <= 50)
        bill = units * 0.50;
    else if (units <= 150)
```

```

    bill = 25 + (units - 50) * 0.75;
else if (units <= 250)
    bill = 100 + (units - 150) * 1.20;
else
    bill = 220 + (units - 250) * 1.50;
printf("Total electricity bill: $%.2f\n", bill);
return 0;
}

```

#### **OUTPUT**

Enter the number of units consumed: 250

Total electricity bill: \$220.00

#### **12. C program to create simple calculator and days of week using switch case.**

```

#include <stdio.h>
int main() {
    int num1, num2, result;
    char operator;
    int day;
    printf("Enter an operator (+, -, *, /): ");
    scanf("%c", &operator);
    printf("Enter two operands: ");
    scanf("%d %d", &num1, &num2);
    switch (operator) {
        case '+':
            result = num1 + num2;
            break;
        case '-':
            result = num1 - num2;
            break;
        case '*':
            result = num1 * num2;
            break;
        case '/':
            result = num1 / num2;
            break;
        default:
            printf("Error! operator is not correct");
            return 0;
    }
    printf("%.1d %.1c %.1d = %.1d", num1, operator, num2, result);
    printf("\nEnter a number between 1 and 7: ");
    scanf("%d", &day);
    switch (day) {

```

```

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("Error! Invalid number entered");
}
return 0;
}

```

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

```

#include <stdio.h>
int main() {
    char ch;
    printf("Enter a character: ");
    scanf("%c", &ch);
    switch (ch) {
        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.", ch);
            break;
        default:
            printf("%c is a consonant.", ch);
    }
    return 0;
}

```

**OUTPUT**

Enter a character: g

g 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("%d is positive.", num);
            break;
        case 0:
            switch (num == 0) {
                case 1:
                    printf("%d is zero.", num);
                    break;
                case 0:
                    printf("%d is negative.", num);
            }
            break;
    }
    return 0;
}

```

**OUTPUT**

Enter a number: 25

25 is positive.

**15. C program to check weather a triangle is equilateral, isosceles or scalene.**

```

#include <stdio.h>
int main() {
    int side1, side2, side3;
    printf("Enter the sides of the triangle: ");
    scanf("%d %d %d", &side1, &side2, &side3);

```

```

switch (side1 == side2) {
    case 1:
        switch (side2 == side3) {
            case 1:
                printf("The triangle is equilateral.");
                break;
            case 0:
                printf("The triangle is isosceles.");
            }
            break;
    case 0:
        switch (side2 == side3) {
            case 1:
                printf("The triangle is isosceles.");
                break;
            case 0:
                printf("The triangle is scalene.");
            }
        }
    return 0;
}

```

**OUTPUT**

Enter the sides of the triangle: 45

60

60

The triangle is isosceles.

**16. C program to print all natural number and sum of it from 1 to n.**

```

#include <stdio.h>
int main(void) {
    int n, i, sum = 0;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    for (i = 1; i <= n; ++i) {
        printf("%d ", i);
        sum += i;
    }
    printf("\nSum = %d", sum);
    return 0;
}

```

**OUTPUT**

Enter a positive integer: 100

```
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 57 58 59 60 61 62 63 64 65 66  
67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97  
98 99 100  
Sum = 5050
```

**17. C program to print all even number and sum of it from 1 to n.**

```
#include <stdio.h>  
int main(void) {  
    int n, i, sum = 0;  
    printf("Enter a positive integer: ");  
    scanf("%d", &n);  
    for (i = 2; i <= n; i += 2) {  
        printf("%d ", i);  
        sum += i;  
    }  
    printf("\nSum = %d", sum);  
    return 0;  
}
```

**OUTPUT**

```
Enter a positive integer: 60  
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  
Sum = 930
```

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

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

**OUTPUT**

```
Enter a number: 17  
17 x 1 = 17  
17 x 2 = 34  
17 x 3 = 51  
17 x 4 = 68  
17 x 5 = 85  
17 x 6 = 102  
17 x 7 = 119
```

$17 \times 8 = 136$   
 $17 \times 9 = 153$   
 $17 \times 10 = 170$

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

```
#include <stdio.h>
int main(void) {
    int n, i, fact = 1;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    for (i = 1; i <= n; ++i) {
        fact *= i;
    }
    printf("Factorial of %d = %d", n, fact);
    return 0;
}
```

**OUTPUT**

Enter a positive integer: 5  
Factorial of 5 = 120

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

```
#include <stdio.h>
int main(void) {
    int n, reversed = 0, remainder, original;
    printf("Enter an integer: ");
    scanf("%d", &n);
    original = n;
    while (n != 0) {
        remainder = n % 10;
        reversed = reversed * 10 + remainder;
        n /= 10;
    }
    if (original == reversed) {
        printf("%d is a palindrome.", original);
    } else {
        printf("%d is not a palindrome.", original);
    }
    return 0;
}
```

**OUTPUT**

Enter an integer: 121  
121 is a palindrome.

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

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

```

int main(void) {
    int n, digit, count[10] = {0};
    printf("Enter a number: ");
    scanf("%d", &n);
    while (n > 0) {
        digit = n % 10;
        count[digit]++;
        n /= 10;
    }
    printf("Digit:\t\t");
    for (int i = 0; i < 10; i++) {
        printf("%d\t", i);
    }
    printf("\nFrequency:\t");
    for (int i = 0; i < 10; i++) {
        printf("%d\t", count[i]);
    }
    return 0;
}

```

**OUTPUT**

Enter a number: 987456321

|            |   |   |   |   |   |   |   |   |   |   |
|------------|---|---|---|---|---|---|---|---|---|---|
| Digit:     | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Frequency: | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

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

```

#include <stdio.h>
int main(void) {
    int n1, n2, i, gcd, lcm;
    printf("Enter two positive integers: ");
    scanf("%d %d", &n1, &n2);
    for (i = 1; i <= n1 && i <= n2; ++i) {
        if (n1 % i == 0 && n2 % i == 0) {
            gcd = i;
        }
    }
    lcm = (n1 * n2) / gcd;
    printf("GCD of %d and %d is %d\n", n1, n2, gcd);
    printf("LCM of %d and %d is %d\n", n1, n2, lcm);
    return 0;
}

```

**OUTPUT**

Enter two positive integers: 16

GCD of 16 and 72 is 8

LCM of 16 and 72 is 144

**23. C program to print all prime number between 1 to n.**

```
#include <stdio.h>
#include <stdbool.h>
int main(void) {
    int n, i, j;
    bool is_prime;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("All prime numbers between 1 and %d are:\n", n);
    for (i = 2; i <= n; ++i) {
        is_prime = true;
        for (j = 2; j <= i / 2; ++j) {
            if (i % j == 0) {
                is_prime = false;
                break;
            }
        }
        if (is_prime) {
            printf("%d ", i);
        }
    }
    return 0;
}
```

**OUTPUT**

Enter a positive integer: 591

All prime numbers between 1 and 591 are:

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127  
131 137 139 149 151 157 163 167 173 179 181 191 193 197 199 211 223 227 229 233 239 241  
251 257 263 269 271 277 281 283 293 307 311 313 317 331 337 347 349 353 359 367 373 379  
383 389 397 401 409 419 421 431 433 439 443 449 457 461 463 467 479 487 491 499 503 509  
521 523 541 547 557 563 569 571 577 587

**24. C program to print all strong number between 1 to n.**

```
#include <stdio.h>
#include <stdbool.h>
int main(void) {
    int n, i, j, last_digit, sum;
    bool is_strong;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("All strong numbers between 1 and %d are:\n", n);
```

```

for (i = 1; i <= n; ++i) {
    sum = 0;
    j = i;
    is_strong = true;
    while (j > 0) {
        last_digit = j % 10;
        j /= 10;
        int fact = 1;
        for (int k = 1; k <= last_digit; ++k) {
            fact *= k;
        }
        sum += fact;
    }

    if (sum == i) {
        printf("%d ", i);
    }
}
return 0;
}

```

**OUTPUT**

Enter a positive integer: 789

All strong numbers between 1 and 789 are:

1 2 145

**25. C program to print Fibonacci series upto n terms.**

```

#include <stdio.h>
int main(void) {
    int n, i, first = 0, second = 1, next;
    printf("Enter the number of terms: ");
    scanf("%d", &n);
    printf("Fibonacci series: ");
    for (i = 1; i <= n; ++i) {
        printf("%d, ", first);
        next = first + second;
        first = second;
        second = next;
    }
}

```

**OUTPUT**

Enter the number of terms: 20

Fibonacci series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181,

**26. C program to print Armstrong number from 1 to n and check a given number is armstrong number or not.**

```
#include <stdio.h>
#include <math.h>
int main(void) {
    int n, i, num, last_digit, digits, sum, original;
    bool is_armstrong;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("All Armstrong numbers between 1 and %d are:\n", n);
    for (i = 1; i <= n; ++i) {
        sum = 0;
        num = i;
        original = num;
        digits = (int)log10(num) + 1;
        while (num > 0) {
            last_digit = num % 10;
            sum += pow(last_digit, digits);
            num /= 10;
        }
        if (original == sum) {
            printf("%d ", original);
        }
    }
    printf("\nEnter a number to check if it is an Armstrong number: ");
    scanf("%d", &num);
    original = num;
    sum = 0;
    digits = (int)log10(num) + 1;
    while (num > 0) {
        last_digit = num % 10;
        sum += pow(last_digit, digits);
        num /= 10;
    }
    is_armstrong = (original == sum);
    printf("%d is ", original);
    if (is_armstrong) {
        printf("an Armstrong number.");
    } else {
        printf("not an Armstrong number.");
    }
    return 0;
}
```

```
}
```

#### OUTPUT

Enter a positive integer: 1000

All Armstrong numbers between 1 and 1000 are:

1 2 3 4 5 6 7 8 9 153 370 371 407

Enter a number to check if it is an Armstrong number: 152

152 is not an Armstrong number.

#### 27. C program to print all perfect number between 1 to n and check a given number is perfect number or not.

```
#include <stdio.h>
int main(void) {
    int n, i, j, sum, num;
    bool is_perfect;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("All perfect numbers between 1 and %d are:\n", n);
    for (i = 2; i <= n; ++i) {
        sum = 0;
        for (j = 1; j < i; ++j) {
            if (i % j == 0) {
                sum += j;
            }
        }
        if (sum == i) {
            printf("%d ", i);
        }
    }
    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;
        }
    }
    is_perfect = (sum == num);
    printf("%d is ", num);
    if (is_perfect) {
        printf("a perfect number.");
    } else {
        printf("not a perfect number.");
```

```
    }
    return 0;
}
```

**OUTPUT**

```
Enter a positive integer: 999
All perfect numbers between 1 and 999 are:
6 28 496
```

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

```
#include <stdio.h>
int main(void) {
    int base, exponent, result = 1;
    printf("Enter base: ");
    scanf("%d", &base);
    printf("Enter exponent: ");
    scanf("%d", &exponent);
    for (int i = 1; i <= exponent; ++i) {
        result *= base;
    }
    printf("%d to the power of %d is %d\n", base, exponent, result);
    return 0;
}
```

**OUTPUT**

```
Enter base: 10
Enter exponent: 3
10 to the power of 3 is 1000
```

**29. C program to print ASCII value s of all characters.**

```
#include <stdio.h>
int main(void) {
    printf("ASCII values of all characters:\n");
    for (int i = 0; i < 256; ++i) {
        printf("%d: %c\n", i, i);
    }
    return 0;
}
```

**OUTPUT**

```
ASCII values of all characters:
0:
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: 0  
49: 1  
50: 2  
51: 3  
52: 4  
53: 5  
54: 6  
55: 7  
56: 8  
57: 9  
58: :  
59: ;  
60: <  
61: =  
62: >  
63: ?  
64: @  
65: A  
66: B  
67: C  
68: D  
69: E  
70: F  
71: G  
72: H  
73: I  
74: J  
75: K  
76: L  
77: M  
78: N  
79: O  
80: P  
81: Q  
82: R  
83: S  
84: T  
85: U  
86: V

87: W  
88: X  
89: Y  
90: Z  
91: [  
92: \  
93: ]  
94: ^  
95: \_  
96: `  
97: a  
98: b  
99: c  
100: d  
101: e  
102: f  
103: g  
104: h  
105: i  
106: j  
107: k  
108: l  
109: m  
110: n  
111: o  
112: p  
113: q  
114: r  
115: s  
116: t  
117: u  
118: v  
119: w  
120: x  
121: y  
122: z  
123: {  
124: |  
125: }  
126: ~  
127: □  
128: Ç

129: ü

130: é

131: â

132: ä

133: à

134: å

135: ç

136: ê

137: ë

138: è

139: ī

140: ī

141: ï

142: Ä

143: Å

144: É

145: æ

146: Æ

147: ô

148: ö

149: ò

150: û

151: ù

152: ÿ

153: Ö

154: Ü

155: ¢

156: £

157: ¥

158: Pts

159: f

160: á

161: í

162: ó

163: ú

164: ñ

165: Ñ

166: ª

167: º

168: ¿

169: –

170: –

171:  $\frac{1}{2}$

172:  $\frac{1}{4}$

173: i

174: «

175: »

176: 

177: 

178: 

179: |

180: -

181: =

182: ||

183: ¶

184: 〽

185: 〽

186: ||

187: 〽

188: 〽

189: 〽

190: 〽

191: 〽

192: L

193: ⊥

194: T

195: -

196: -

197: +

198: =

199: ||

200: 〽

201: 〽

202: 〽

203: 〽

204: 〽

205: =

206: +

207: =

208: 〽

209: 〽

210: 〽

211: 〽

212: L

213:  $\mathbb{F}$

214:  $\mathbb{R}$

215:  $\mathbb{H}$

216:  $\mathbb{P}$

217:  $\mathbb{J}$

218:  $\Gamma$

219:  $\blacksquare$

220:  $\blacksquare$

221:  $\blacksquare$

222:  $\blacksquare$

223:  $\blacksquare$

224:  $\alpha$

225:  $\beta$

226:  $\Gamma$

227:  $\pi$

228:  $\Sigma$

229:  $\sigma$

230:  $\mu$

231:  $\tau$

232:  $\Phi$

233:  $\Theta$

234:  $\Omega$

235:  $\delta$

236:  $\infty$

237:  $\phi$

238:  $\varepsilon$

239:  $\cap$

240:  $\equiv$

241:  $\pm$

242:  $\geq$

243:  $\leq$

244:  $\lceil$

245:  $\rfloor$

246:  $\div$

247:  $\approx$

248:  $^\circ$

249:  $\cdot$

250:  $\cdot$

251:  $\vee$

252:  $^n$

253:  $^2$

254:  $\blacksquare$

255:

**30. C program to print pascal triangle upto 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;
}
```

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

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

```

int arr[100], size, i, sum = 0;
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 = sum + arr[i];
}
printf("Sum of all elements of array = %d", sum);
return 0;
}

```

#### **OUTPUT**

Enter size of the array: 5

Enter elements of the array: 6

5

3

9

4

Sum of all elements of array = 27

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

```

#include <stdio.h>
int main()
{
    int source[100], target[100], size, i;
    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]);
    }
    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;
}

```

```
}
```

#### OUTPUT

```
Enter size of the array: 10
```

```
Enter elements of the source array: 36
```

```
69
```

```
45
```

```
98
```

```
47
```

```
21
```

```
45
```

```
35
```

```
25
```

```
84
```

```
Elements of target array: 36 69 45 98 47 21 45 35 25 84
```

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

```
#include <stdio.h>
int main()
{
    int arr[100], size, i, pos, element;
    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 you want to insert an element: ");
    scanf("%d", &pos);
    printf("Enter the element you want to insert: ");
    scanf("%d", &element);
    for(i=size-1; i>=pos; i--)
    {
        arr[i+1] = arr[i];
    }
    arr[pos] = element;
    printf("Array after insertion: ");
    for(i=0; i<=size; i++)
    {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

**OUTPUT**

```
Enter size of the array: 5
Enter elements of the array: 4
9
8
2
6
```

```
Enter the position where you want to insert an element: 2
Enter the element you want to insert: 47
Array after insertion: 4 9 47 8 2 6
```

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

```
#include <stdio.h>
int main()
{
    int arr[100], size, i, pos;
    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 of the element you want to delete: ");
    scanf("%d", &pos);
    for(i=pos-1; i<size-1; i++)
    {
        arr[i] = arr[i+1];
    }
    printf("Array after deleting an element: ");
    for(i=0; i<size-1; i++)
    {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

**OUTPUT**

```
Enter size of the array: 4
Enter elements of the array: 3
7
1
9
```

```
Enter the position of the element you want to delete: 3
```

Array after deleting an element: 3 7 9

**35. C program to search element in array using linear search.**

```
#include <stdio.h>
#define ARRAY_SIZE 10
int main(void) {
    int array[ARRAY_SIZE] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int search_key = 5;
    int found = 0;
    int i;
    for (i = 0; i < ARRAY_SIZE; i++) {
        if (array[i] == search_key) {
            found = 1;
            break;
        }
    }
    if (found) {
        printf("The search key was found at index %d\n", i);
    } else {
        printf("The search key was not found in the array\n");
    }
    return 0;
}
```

**OUTPUT**

The search key was found at index 4

**36. C program to find second largest number and sorting using bubble sort in an array.**

```
#include <stdio.h>
#define ARRAY_SIZE 10
void bubble_sort(int array[], int size) {
    int i, j;
    for (i = 0; i < size - 1; i++) {
        for (j = 0; j < size - i - 1; j++) {
            if (array[j] > array[j + 1]) {
                int temp = array[j];
                array[j] = array[j + 1];
                array[j + 1] = temp;
            }
        }
    }
}
int main(void) {
    int array[ARRAY_SIZE] = {9, 8, 7, 6, 5, 4, 3, 2, 1, 0};
    int second_largest;
```

```
bubble_sort(array, ARRAY_SIZE);
second_largest = array[ARRAY_SIZE - 2];
printf("The second largest number is %d\n", second_largest);
return 0;
}
```

**OUTPUT**

The second largest number is 8

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

```
#include <stdio.h>
#define ARRAY_SIZE 10
int main(void) {
    int array[ARRAY_SIZE] = {1, 2, 3, 3, 4, 4, 4, 5, 5, 5};
    int i, j;
    int count = 0;
    for (i = 0; i < ARRAY_SIZE; i++) {
        for (j = i + 1; j < ARRAY_SIZE; j++) {
            if (array[i] == array[j]) {
                count++;
                break;
            }
        }
    }
    printf("The total number of duplicate elements is %d\n", count);
    return 0;
}
```

**OUTPUT**

The total number of duplicate elements is 5

**38. C program to perform scalar matrix multiplication.**

```
#include <stdio.h>
#define ROWS 3
#define COLUMNS 3
int main(void) {
    int matrix[ROWS][COLUMNS] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int scalar = 2;
    int i, j;
    for (i = 0; i < ROWS; i++) {
        for (j = 0; j < COLUMNS; j++) {
            matrix[i][j] *= scalar;
        }
    }
    for (i = 0; i < ROWS; i++) {
        for (j = 0; j < COLUMNS; j++) {
```

```
    printf("%d ", matrix[i][j]);
}
printf("\n");
}
return 0;
}
```

**OUTPUT**

```
2 4 6
8 10 12
14 16 18
```

**39. C program to find sum of main diagonal element of a matrix.**

```
#include <stdio.h>
#define ROWS 3
#define COLUMNS 3
int main(void) {
    int matrix[ROWS][COLUMNS] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int i;
    int sum = 0;
    for (i = 0; i < ROWS; i++) {
        sum += matrix[i][i];
    }
    printf("The sum of the main diagonal elements is %d\n", sum);
    return 0;
}
```

**OUTPUT**

```
The sum of the main diagonal elements is 15
```

**40. C program to check sparse and transpose matrix.**

```
#include <stdio.h>
#define ROWS 5
#define COLUMNS 5
int main(void) {
    int matrix[ROWS][COLUMNS] = {{0, 1, 0, 0, 0},
                                {0, 0, 2, 0, 0},
                                {0, 0, 0, 3, 0},
                                {0, 0, 0, 0, 4},
                                {0, 0, 0, 0, 0}};
    int i, j;
    int is_sparse = 1;
    int is_transpose = 1;
    for (i = 0; i < ROWS; i++) {
        for (j = 0; j < COLUMNS; j++) {
            if (matrix[i][j] != 0) {
```

```

        is_sparse = 0;
        break;
    }
}
}
for (i = 0; i < ROWS; i++) {
    for (j = 0; j < COLUMNS; j++) {
        if (matrix[i][j] != matrix[j][i]) {
            is_transpose = 0;
            break;
        }
    }
}
if (is_sparse && is_transpose) {
    printf("The matrix is both sparse and transpose\n");
} else {
    printf("The matrix is not both sparse and transpose\n");
}
return 0;
}

```

#### **OUTPUT**

The matrix is not both sparse and transpose

#### **41. C program to check weather a matrix is identify matrix or not.**

```

#include <stdio.h>
#define ROWS 3
#define COLUMNS 3
int main(void) {
    int matrix[ROWS][COLUMNS] = {{1, 0, 0},
                                {0, 1, 0},
                                {0, 0, 1}};
    int i, j;
    int is_identity = 1;
    for (i = 0; i < ROWS; i++) {
        for (j = 0; j < COLUMNS; j++) {
            if (i == j && matrix[i][j] != 1) {
                is_identity = 0;
                break;
            } else if (i != j && matrix[i][j] != 0) {
                is_identity = 0;
                break;
            }
        }
    }
}
```

```

    }
    if (is_identity) {
        printf("The matrix is an identity matrix\n");
    } else {
        printf("The matrix is not an identity matrix\n");
    }
    return 0;
}

```

### **OUTPUT**

The matrix is an identity matrix

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

```

#include <stdio.h>
#define ARRAY_SIZE_1 5
#define ARRAY_SIZE_2 8
#define MERGED_ARRAY_SIZE (ARRAY_SIZE_1 + ARRAY_SIZE_2)
int main(void) {
    int array_1[ARRAY_SIZE_1] = {1, 3, 5, 7, 9};
    int array_2[ARRAY_SIZE_2] = {2, 4, 6, 8, 10, 12, 14, 16};
    int merged_array[MERGED_ARRAY_SIZE];
    int i = 0, j = 0, k = 0;
    while (i < ARRAY_SIZE_1 && j < ARRAY_SIZE_2) {
        if (array_1[i] < array_2[j]) {
            merged_array[k] = array_1[i];
            i++;
        } else {
            merged_array[k] = array_2[j];
            j++;
        }
        k++;
    }
    while (i < ARRAY_SIZE_1) {
        merged_array[k] = array_1[i];
        i++;
        k++;
    }
    while (j < ARRAY_SIZE_2) {
        merged_array[k] = array_2[j];
        j++;
        k++;
    }
    for (i = 0; i < MERGED_ARRAY_SIZE; i++) {
        printf("%d ", merged_array[i]);
    }
}

```

```

    }
    printf("\n");
    return 0;
}

```

**OUTPUT**

1 2 3 4 5 6 7 8 9 10 12 14 16

**43. C program to perform all operation of string.**

```

#include<stdio.h>
#include<conio.h>
void main(){
Char string1[25],string2[25];
int l;
Clrscr();
printf("***** performing string length *****\n");
printf("enter only one string \n");
scanf("%s",string1);
l = strlen(string1);
printf("the string length is %d\n\n",l);
printf("***** performing string concatenation ****\n");
printf("enter two strings\n");
scanf("%s%s",string1,string2);
printf("the concatenated string is %s\n\n",strcat(string1,string2));
printf("***** performing string compare *****\n");
printf("enter two strings \n");
scanf("%s%s",string1,string2);

if(strcmp(string1,string2) == 0)
printf("strings are equal\n");

else
printf("strings are not equal\n");
printf("**** performing string copy ****\n");
printf("enter the two strings\n");
scanf("%d%d",string1,string2);
printf("the first string is %s and second string is %s\n",string1,string2);
strcpy(string1,string2);
printf("the first string is %s and second string is %s\n",string1,string2);
return 0;
}

```

**44. C program to check weather a string is palindrome or not without compare function of string.**

```

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

```

```

int main()
{
    char str[] = { "abbba" };
    int l = 0;
    int h = strlen(str) - 1;
    while (h > l) {
        if (str[l++] != str[h--]) {
            printf("%s is not a palindrome\n", str);
            return 0;
        }
    }
    printf("%s is a palindrome\n", str);
    return 0;
}

```

**OUTPUT**

abbba is a palindrome

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

```

#include <stdio.h>
int main() {
    char str[1000], ch;
    int count = 0;
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    printf("Enter a character to find its frequency: ");
    scanf("%c", &ch);
    for (int i = 0; str[i] != '\0'; ++i) {
        if (ch == str[i])
            ++count;
    }
    printf("Frequency of %c = %d", ch, count);
    return 0;
}

```

**OUTPUT**

Enter a string: My name is Bhuvan Agrawal of section s studing in GLA University

Enter a character to find its frequency: s

Frequency of s = 5

**46. C program to find diameter, circumference and area of circle using functions.**

```

#include <stdio.h>
#include <math.h>
double getDiameter(double radius);
double getCircumference(double radius);

```

```

double getArea(double radius);
int main()
{
    float radius, dia, circ, area;
    printf("Enter radius of circle: ");
    scanf("%f", &radius);
    dia = getDiameter(radius);
    circ = getCircumference(radius);
    area = getArea(radius);
    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;
}
double getDiameter(double radius)
{
    return (2 * radius);
}
double getCircumference(double radius)
{
    return (2 * M_PI * radius);
}
double getArea(double radius)
{
    return (M_PI * radius * radius);
}

```

#### **OUTPUT**

```

Enter radius of circle: 21
Diameter of the circle = 42.00 units
Circumference of the circle = 131.95 units
Area of the circle = 1385.44 sq. units

```

#### **47. C program to check prime, Armstrong and perfect numbers using functions.**

```

#include <stdio.h>
#include <math.h>
int isPrime(int num);
int isArmstrong(int num);
int isPerfect(int num);
int main()
{
    int num;
    printf("Enter any number: ");
    scanf("%d", &num);

```

```

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

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

if(isPerfect(num))
{
    printf("%d is Perfect number.\n", num);
}
else
{
    printf("%d is not Perfect number.\n", num);
}
return 0;
}

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

int isArmstrong(int num)
{
    int lastDigit, sum, originalNum, digits;

```

```

sum = 0;
originalNum = num;
    digits = (int) log10(num) + 1;
    while(num > 0)
{
    lastDigit = num % 10;
        sum = sum + round(pow(lastDigit, digits));
        num = num / 10;
}
return (originalNum == sum);
}

int isPerfect(int num)
{
    int i, sum, n;
    sum = 0;
    n = num;
    for(i=1; i<n; i++)
    {
        if(n%i == 0)
        {
            sum += i;
        }
    }
    return (num == sum);
}

```

#### **OUTPUT**

Enter any number: 7  
 7 is Prime number.  
 7 is Armstrong number.  
 7 is not Perfect number.

#### **48. C program to add two numbers 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 entered numbers = %d\n",sum);
    return 0;
}

```

**OUTPUT**

Enter two integers to add

5

10

Sum of entered numbers = 15

**49. C program to swap 2 numbers using call by value and 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;
}
```

**OUTPUT**

Enter the value of x and y

50

100

Before Swapping

x = 50

y = 100

After Swapping

x = 100

y = 50

**50. C program to copy an array to another array and reverse an array using pointers.**

```
#include <stdio.h>
#define MAX_SIZE 100
void printArr(int *arr, int size);
int main()
{
    int arr[MAX_SIZE];
```

```

int size;
int *left = arr;
int *right;
printf("Enter size of array: ");
scanf("%d", &size);
right = &arr[size - 1];
printf("Enter elements in array: ");
while(left <= right)
{
    scanf("%d", left++);
}
printf("\nArray before reverse: ");
printArr(arr, size);
left = arr;
while(left < right)
{
    *left ^= *right;
    *right ^= *left;
    *left ^= *right;
    left++;
    right--;
}
printf("\nArray after reverse: ");
printArr(arr, size);
return 0;
}
void printArr(int * arr, int size)
{
    int * arrEnd = (arr + size - 1);
    while(arr <= arrEnd)
    {
        printf("%d, ", *arr);
        arr++;
    }
}

OUTPUT
Enter size of array: 4
Enter elements in array: 82
59
41
37

```

Array before reverse: 82, 59, 41, 37

Array after reverse: 37, 41, 59, 82

## Pattern Programming

### 1. Square Star Pattern

```
#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;
}
```

#### OUTPUT

Enter number of rows: 10

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

### 2. Right Triangle Square Pattern

```
#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;
}

OUTPUT
Enter value of n: 10
*
**
***
****
*****
*****
*****
*****
*****
*****
```

### **3. Mirrored Right Triangle Star Pattern**

```
#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;
}
```

## OUTPUT

Enter number of rows: 10

\*  
\*\*  
\*\*\*  
\* \* \* \*  
\* \* \* \* \*  
\* \* \* \* \* \*  
\* \* \* \* \* \* \*  
\* \* \* \* \* \* \* \*

#### 4. Square Number Pattern 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;
}
```

## OUTPUT

Enter number of rows: 4

Enter number of columns: 5

11111

11111

11111

11111

11111

### 5. Square Number Pattern 2

```
#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++)
        {
            if(i%2 == 1)
            {
                printf("1");
            }
            else
            {
                printf("0");
            }
        }
        printf("\n");
    }
    return 0;
}
```

### OUTPUT

Enter number of rows: 5

Enter number of columns: 6

111111

000000

111111

000000

111111