OPERATORS AND EXPRESSIONS

SARTHAK SANAY

(1) AIM:-

To write a program in C which checks whether a year given by the user is a leap year or not by using logical operators.

CODE:-

```c
#include <stdio.h>
int main()
{
    int year;
    printf("Enter year: ");
    scanf("%d", &year);
    if (((year % 4 == 0) && (year % 100!= 0)) || (year%400 == 0))
        printf("%d is a leap year!", year);
    else
        printf("%d is not a leap year!", year);
    return 0;
}
```

OUTPUT SCREEN:-

```
Enter year: 2024
2024 is a leap year!
```

```
Enter year: 2023
2023 is not a leap year!
```

(2) AIM:-

To write a program in C to calculate the area of a triangle using Heron's Formula.

CODE:-

```c
#include <math.h>
#include <stdio.h>
int main()
{
    int s, a, b, c;
    printf("Enter the value of semi-perimeter (s) : ");
    scanf("%d", &s);
    printf("Enter side a : ");
    scanf("%d", &a);
    printf("Enter side b : ");
    scanf("%d", &b);
    printf("Enter side c : ");
    scanf("%d", &c);
    double x = s*(s-a)*(s-b)*(s-c);
    float area = sqrt(x);
    printf("Area of the triangle is %f", area);
    return 0;
}
```

OUTPUT SCREEN:-

```
Output
/tmp/g9bLqKY57R.o
Enter the value of semi-perimeter (s) : 24
Enter side a : 10
Enter side b : 17
Enter side c : 21
Area of the triangle is 84.000000A
```

(3) AIM:-

To write programs in C to demonstrate the usage of arithmetic, relational, logical, and bitwise operators.

CODE 1:- (Arithmetic Operators)

```c
// Program in C to demonstrate the use of Arithmetic operators
#include <stdio.h>
int main()
{
    int a=20, b=10;
    printf("%d + %d = %d\n", a, b, a+b); // Addition
    printf("%d - %d = %d\n", a, b, a-b); // Subtraction
    printf("%d * %d = %d\n", a, b, a*b); // Multiplication
    printf("%d / %d = %d\n", a, b, a/b); // Division
    return 0;
}
```

OUTPUT SCREEN 1:-

```
Output
/tmp/rgQmLf3XG6.o
20 + 10 = 30
20 - 10 = 10
20 * 10 = 200
20 / 10 = 2
```

CODE 2:- (Relational Operators)

```c
// Program in C to demonstrate the use of Relational operators
#include <stdio.h>
int main()
{
    int a = 10, b = 20;
    // Using ternary operators to return the output
    printf("%d==%d is %s\n", a, b, a==b? "true" : "false"); // equal to
    printf("%d!=%d is %s\n", a, b, a!=b ? "true" : "false"); // not equal
to
    printf("%d>%d is %s\n", a, b, a>b ? "true" : "false"); // greater than
    printf("%d<%d is %s\n", a, b, a<b ? "true" : "false"); // less than
    printf("%d>=%d is %s\n", a, b, a>=b ? "true" : "false"); // greater
than or equal to
    printf("%d<=%d is %s\n", a, b, a<=b ? "true":"false"); // less than or
equal to
    return 0;
}
```

OUTPUT SCREEN 2:-

```
Output
/tmp/rgQmLf3XG6.o
10==20 is false
10!=20 is true
10>20 is false
10<20 is true
10>=20 is false
10<=20 is true
```

CODE 3:- (Logical Operators)

```c
// Program in C to demonstrate the use of Logical operators
#include <stdio.h>
int main()
{
    int a=1, b=0;
    // Using ternary operators to return the output

    // Logical AND
    printf("(%d && %d) is %s\n", a, b, a&&b ? "true":"false");
    // Logical OR
    printf("(%d || %d) is %s\n", a, b, a||b ? "true":"false");
    // Logical NOT
    printf("!(%d) is %s\n", a, !a ? "true" : "false");
    printf("!(%d) is %s\n", b, !b ? "true" : "false");
    return 0;
}
```

OUTPUT SCREEN 3:-

```
Output
/tmp/rgQmLf3XG6.o
(1 && 0) is false
(1 || 0) is true
!(1) is false
!(0) is true
```

CODE 4:- (Bitwise Operators)

```c
// Program in C to demonstrate the use of Bitwise operators
#include <stdio.h>
int main()
{
    int a= 5, b= 3;
    printf("Bitwise AND: \t%d & %d = %d\n", a, b, a&b);
    printf("Bitwise OR: \t%d | %d = %d\n", a, b, a|b);
    printf("Bitwise XOR: \t%d ^ %d = %d\n", a, b, a^b);
    printf("Bitwise NOT: \t~%d = %d\n", a, ~a);
    printf("Bitwise NOT: \t~%d = %d\n", b, ~b);
    printf("Left shift: \t%d << 1 = %d\n", a, a<<1);
    printf("Right shift: \t%d >> 1 = %d\n", a, a>>1);
    return 0;
}
```

OUTPUT SCREEN 4:-

```
Output
/tmp/rgQmLf3XG6.o
Bitwise AND:    5 & 3 = 1
Bitwise OR:     5 | 3 = 7
Bitwise XOR:    5 ^ 3 = 6
Bitwise NOT:    ~5 = -6
Bitwise NOT:    ~3 = -4
Left shift:     5 << 1 = 10
Right shift:    5 >> 1 = 2
```

(4) AIM:-

To write a program in C that swaps two numbers using arithmetic and bitwise operators.

CODE 1:- (Swapping using Arithmetic operators)

```c
// Swapping two numbers using Arithmetic operators
#include <stdio.h>
int main()
{
    int a = 5, b = 10;
    printf("Before swapping: \ta=%d   b=%d\n", a, b);
    a= a+b;
    b= a-b;
    a= a-b;
    printf("After swapping: \ta=%d   b=%d\n", a, b);
    return 0;
}
```

OUTPUT SCREEN 1:-

```
Output
/tmp/rgQmLf3XG6.o
Before swapping:    a=5   b=10
After swapping:     a=10   b=5
```

CODE 2:- (Swapping using Bitwise operators)

```c
// Swapping two numbers using Bitwise operators
#include <stdio.h>
int main()
{
    int a = 5, b = 10;
    printf("Before swapping: \ta=%d   b=%d\n", a, b);
    // Swapping using bitwise XOR
    a= a^b;
    b= a^b;
    a= a^b;
    printf("After swapping: \ta=%d   b=%d\n", a, b);
    return 0;
}
```

OUTPUT SCREEN 2:-

```
Output
/tmp/rgQmLf3XG6.o
Before swapping:    a=5   b=10
After swapping:     a=10   b=5
```