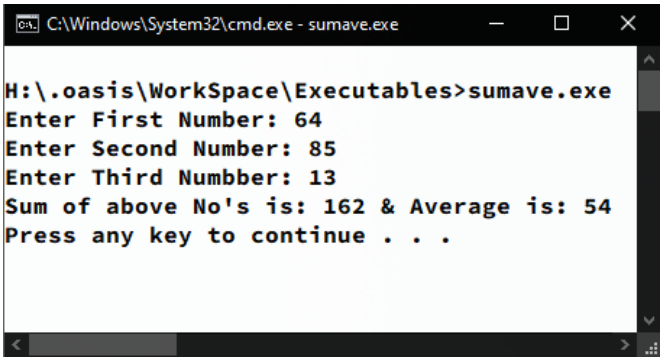
/* C++ Program to Sum & Average of 3 Numbers. */

```
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int num1, num2, num3, sum; float avg;
    cout << "Enter First Number: ";
    cin >> num1;
    cout << "Enter Second Number: ";
    cin >> num2;
    cout << "Enter Third Numbber: ";
    cin >> num3;
    sum = num1 + num2 + num3;
    avg = sum / 3;
    cout << "Sum of above No's is: " << sum << " & Average is: " << avg
<< endl;
    system("pause");
    return 0;
}
```

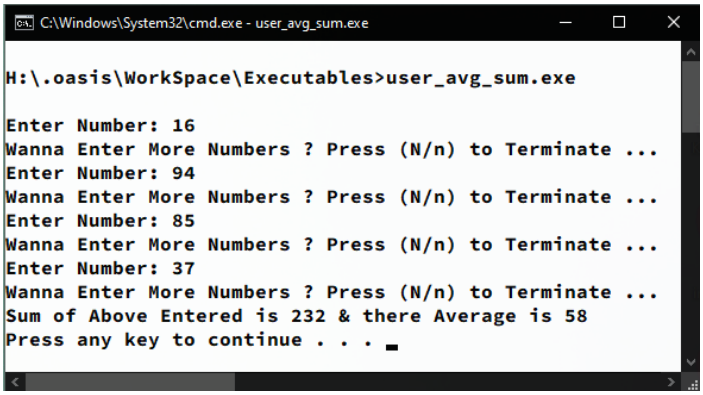
Output:



/* C++ Program to find Sum & Average of N numbers. */

```
#include<iostream>
#include<conio.h>
using namespace std;
int main(int argc, char const *argv[]) {
    float num, sum = 0, avg, count=0;
    char ch;
    do{
        cout << "\nEnter Number: ";
        cin >> num;
        sum += num;
        count++;
        avg = sum / count;
      cout << "Wanna Enter More Numbers ? Press (N/n) to Terminate …";
        ch = getch();
    } while (ch != 'N' && ch != 'n');
cout << "\nSum of Above Entered is " << sum << " & their Average is " << avg << endl;
    system("pause");
    return 0;
}
```

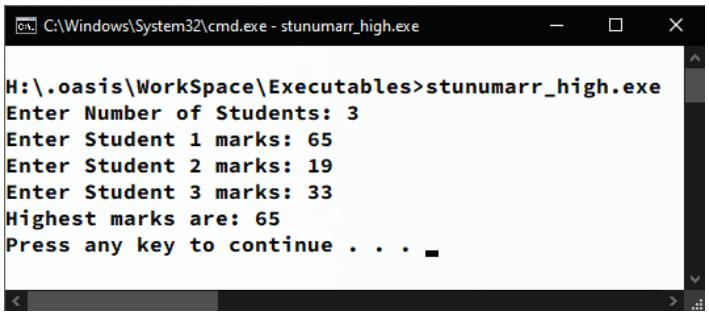
Output:



/* C++ Program to get highest marks among N Students. */

```
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int top = 0, max;
    cout << "Enter Number of Students: ";
    cin >> max;
    int num[max];
    for (int i = 0; i < max; i++) {
        cout << "Enter Student " << i + 1 << " marks: ";
        cin >> num[i];
        if (num[i] > top)
            top = num[i];
    }
    cout << "Highest marks are: " << top << endl;
    system("pause");
    return 0;
}
```

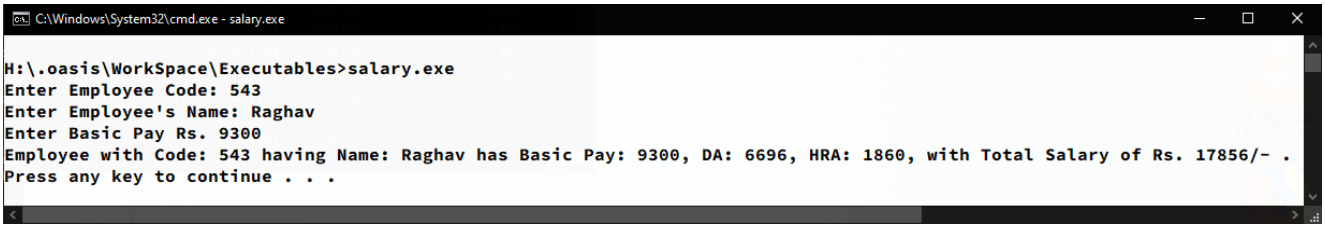
Output:



/* C++ Program to calculate Net Pay of Employee. */

```
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    float bp, da, hra, net, code; string name;
    cout << "Enter Employee Code: ";
    cin >> code;
    cout << "Enter Employee's Name: ";
    cin >> name;
    cout << "Enter Basic Pay Rs. ";
    cin >> bp;
    da = 0.72 * bp;
    hra = 0.20 * bp;
    net = bp + da + hra;
    cout << "Employee with Code: " << code << " having Name: " << name <<
" has Basic Pay: " << bp << ", DA: " << da << ", HRA: " << hra << ", with
Total Salary of Rs. " << net <<"/- ."<< endl;
    system("pause");
    return 0;
}
```

Output:

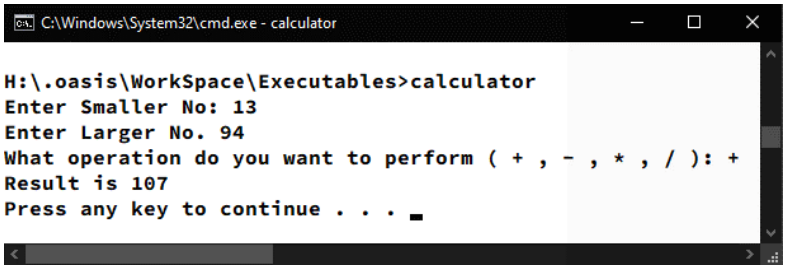


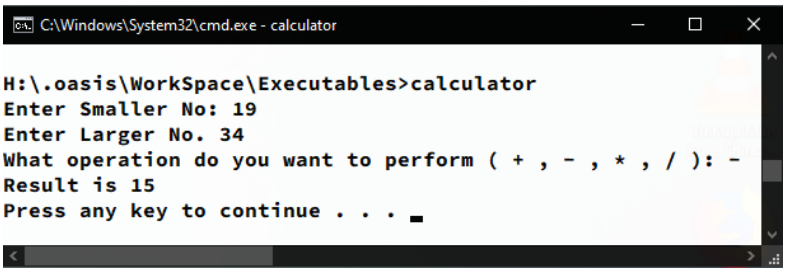
/* C++ Program to Implement a Simple Calculator. */

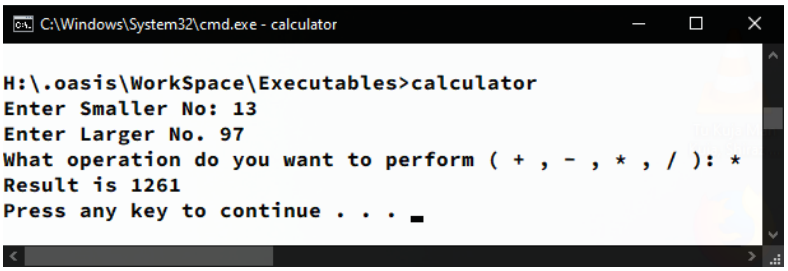
```
#include <iostream>
#include <conio.h>
using namespace std;
int main(int argc, char const *argv[]) {
    float x, y, res; char ch; bool repeat;
    cout << "Enter Smaller No: ";
    cin >> x;
    cout << "Enter Larger No. ";
    cin >> y;
    do {
        repeat = false;
     cout << "What operation do you want to perform ( + , - , * , / ): ";
        cin >> ch;
        switch (ch) {
        case '+':
            res = x + y;
            break;
        case '-':
            res = y - x;
            break;
        case '*':
            res = x * y;
            break;
        case '/':
            res = y / x;
            break;
        default:
            cout << "Wrong Selection! Choose Again !!!\n";
```

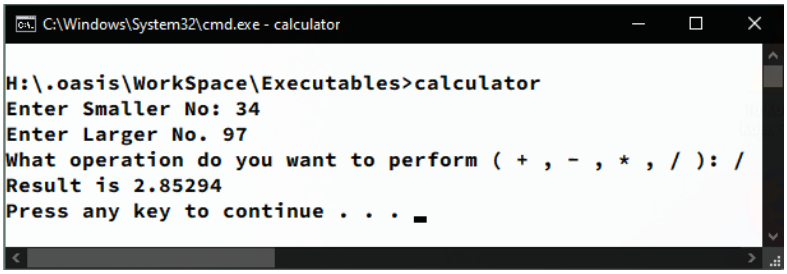
```
            repeat = true;
            break;
        }
    } while (repeat);
    cout << "Result is " << res << endl;
    system("pause");
    return 0;
}
```

Output:





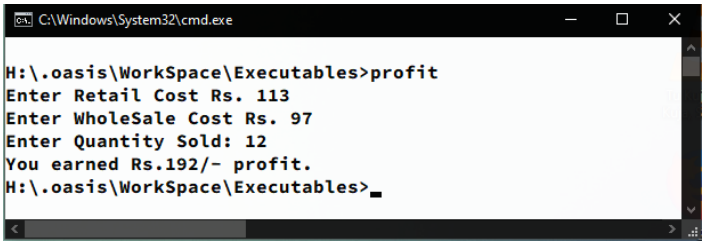




/* C++ Program to Calculate generated Profit. */

```
#include<iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    float quantity, retail_price, wholesale_price, profit;
    cout << "Enter Retail Cost Rs. ";
    cin >> retail_price;
    cout << "Enter WholeSale Cost Rs. ";
    cin >> wholesale_price;
    cout << "Enter Quantity Sold: ";
    cin >> quantity;
    profit = quantity * (retail_price - wholesale_price);
    cout << "You earned Rs." << profit << "/- profit.";
    return 0;
}
```

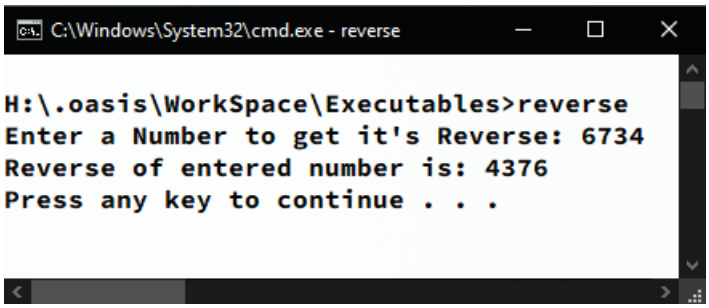
Output:



/* C++ Program to get Reverse of a Number. */

```
#include<iostream>
using namespace std;
int main(int argc, char const *argv[]){
    int num, rev = 0;
    cout << "Enter a Number to get it's Reverse: ";
    cin >> num;
    while(num > 0){
        rev = (rev * 10) + (num % 10);
        num /= 10;
    }
    cout << "Reverse of entered number is: " << rev << endl;
    system("pause");
    return 0;
}
```

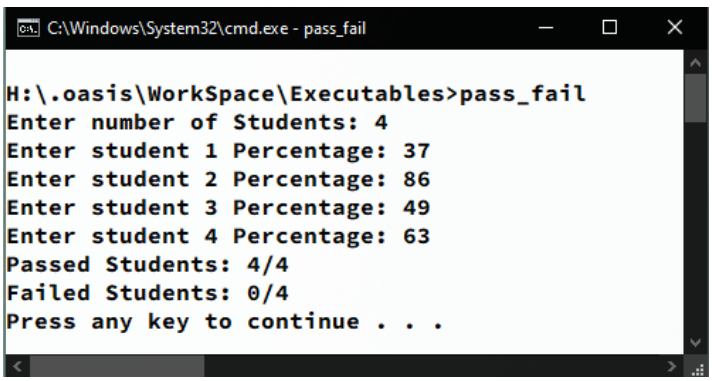
Output:



/* C++ Program to find number of Passed & Failed Students among N Students. */

```
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int students, percentage, pass = 0, fail = 0;
    cout << "Enter number of Students: ";
    cin >> students;
    for (int i = 0; i < students; i++) {
        cout << "Enter student " << i + 1 << " Percentage: ";
        cin >> percentage;
        if (percentage > 100) {
            i--;
            cout << "Percentage can't be greater than 100% :)" << endl;
        }
        else if (percentage >= 33 && percentage <= 100) pass++;
        else fail++;
    }
    cout << "Passed Students: " << pass << "/" << students << endl;
    cout << "Failed Students: " << fail << "/" << students << endl;
    system("pause");
    return 0;
}
```

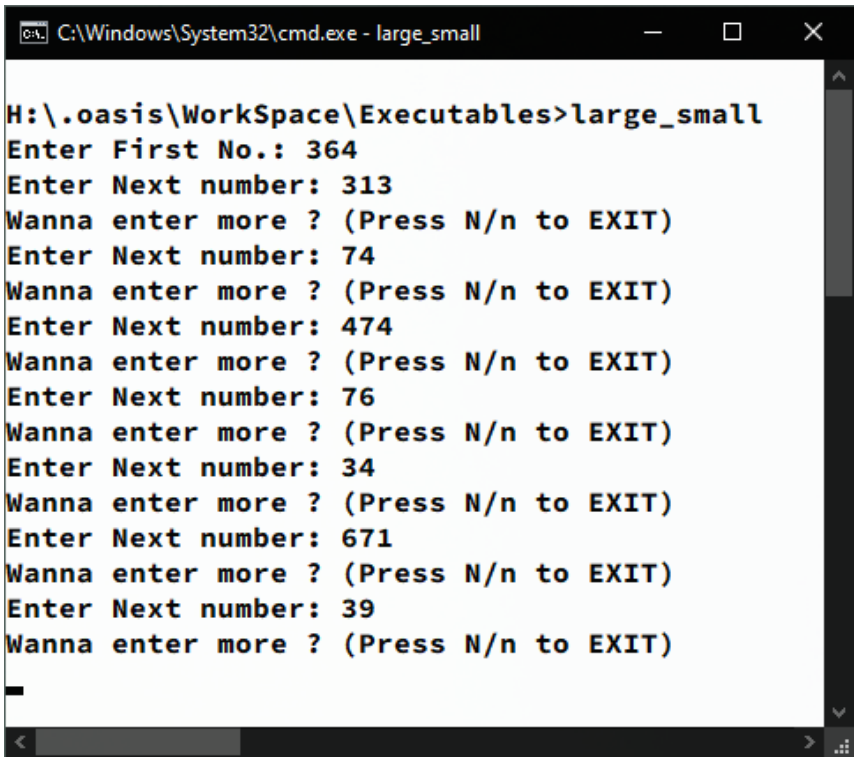
Output:

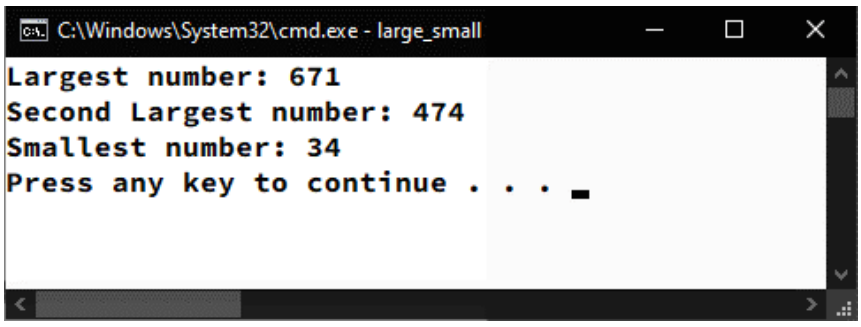


/* C++ Program to find Largest, Second largest & Smallest number among N numbers. */

```
#include <iostream>
#include <conio.h>
using namespace std;
int main(int argc, char *argv[]) {
    int smallest, largest, sLarge, num;
    char ch;
    cout << "Enter First No.: ";
    cin >> num;
    smallest = largest = sLarge = num;
    do {
        cout << "Enter Next number: ";
        cin >> num;
        if (num > largest){
            sLarge = largest;
            largest = num;
        } else if (num < smallest)
            smallest = num;
        cout << "Wanna enter more ? (Press N/n to EXIT)\n";
        ch = getch();
    } while (ch != 'n' && ch != 'N');
    system("cls");
    cout << "Largest number: " << largest << endl;
    cout << "Second Largest number: " << sLarge << endl;
    cout << "Smallest number: " << smallest << endl;
    system("pause");
    return 0;
}
```

Output:

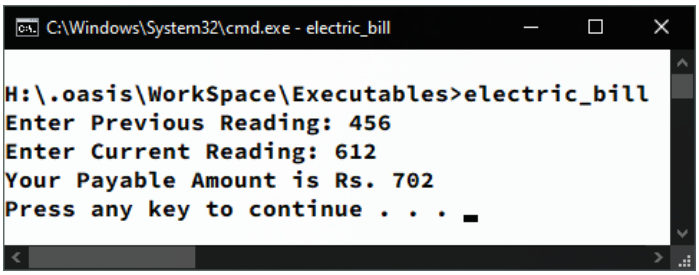

C:\Windows\System32\cmd.exe - large_small
H:\.oasis\WorkSpace\Executables>large_small
Enter First No.: 364
Enter Next number: 313
Wanna enter more ? (Press N/n to EXIT)
Enter Next number: 74
Wanna enter more ? (Press N/n to EXIT)
Enter Next number: 474
Wanna enter more ? (Press N/n to EXIT)
Enter Next number: 76
Wanna enter more ? (Press N/n to EXIT)
Enter Next number: 34
Wanna enter more ? (Press N/n to EXIT)
Enter Next number: 671
Wanna enter more ? (Press N/n to EXIT)
Enter Next number: 39
Wanna enter more ? (Press N/n to EXIT)


C:\Windows\System32\cmd.exe - large_small
Largest number: 671
Second Largest number: 474
Smallest number: 34
Press any key to continue . . .

/* C++ Program to get Electric Bill amount at fixed Rates. */

```
#include<iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    float pre_read, cur_read, units;
    cout << "Enter Previous Reading: ";
    cin >> pre_read;
    cout << "Enter Current Reading: ";
    cin >> cur_read;
    units = cur_read - pre_read;
    if(units < 200){
        cout << "Your Payable Amount is Rs. " << units * 4.50 << endl;
    }else if(units >= 200 && units < 300){
        cout << "Your Payable Amount is Rs. " << units * 5.00 << endl;
    }else if(units >= 300 && units < 400){
        cout << "Your Payable Amount is Rs. " << units * 5.50 << endl;
    }else if(units >= 400){
        cout << "Your Payable Amount is Rs. " << units * 6.00 << endl;
    }
    system("pause");
    return 0;
}
```

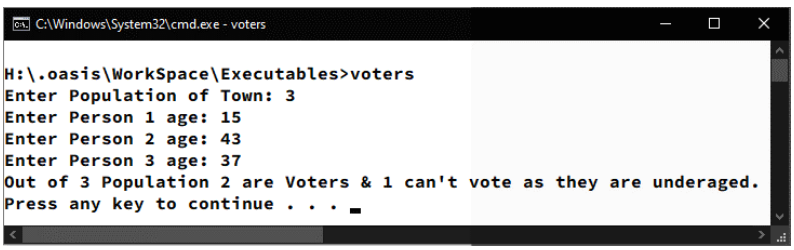
Output:



/* C++ Program to Get No. of Voters in a Town given their Ages. */

```
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int pop, age, i, voters = 0;
    cout << "Enter Population of Town: ";
    cin >> pop;
    i = pop;
    while (i > 0) {
        cout << "Enter Person " << (pop - i) + 1 << " age: ";
        cin >> age;
        if (age >= 18) voters++;
        i--;
    }
    cout << "Out of " << pop << " Population " << voters << " are Voters
& " << pop - voters << " can't vote as they are underaged." << endl;
    system("pause");
    return 0;
}
```

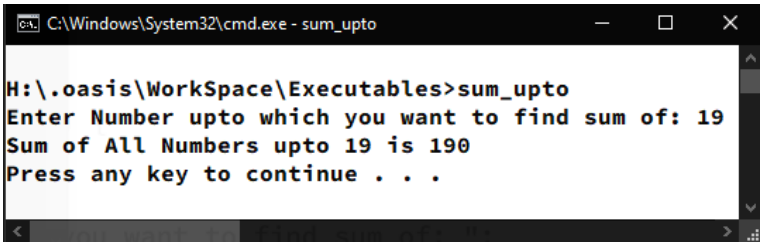
Output:



/* C++ Program to find sum of first N Numbers. */

```
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int num, sum = 0;
    cout << "Enter Number upto which you want to find sum of: ";
    cin >> num;
    for (int i = 1; i <= num; i++) sum += i;
    cout << "Sum of All Numbers upto " << num << " is " << sum << endl;
    system("pause");
    return 0;
}
```

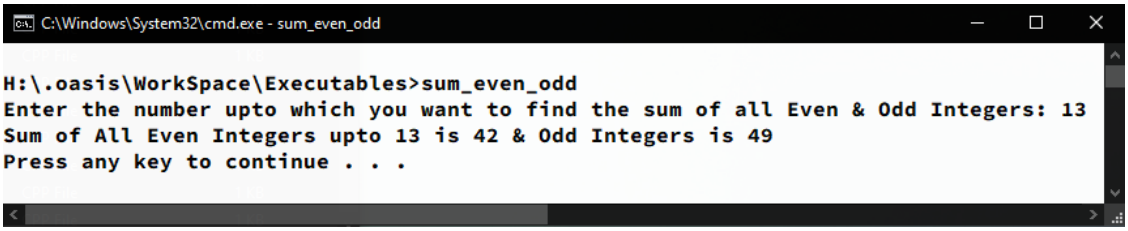
Output:



/* C++ Program to find Sum of All Even & Odd Integers up to N. */

```
#include <iostream>

using namespace std;

int main(int argc, char const *argv[]) {

    int upto, even = 0, odd = 0;

    cout << "Enter the number upto which you want to find the sum of all
Even & Odd Integers: ";

    cin >> upto;

    for (int i = 0; i <= upto; i++) {

        if (i % 2 == 0)

            even += i;

        else

            odd += i;

    }

    cout << "Sum of All Even Integers upto " << upto << " is " << even <<
" & Odd Integers is " << odd << endl;

    system("pause");

    return 0;

}
```

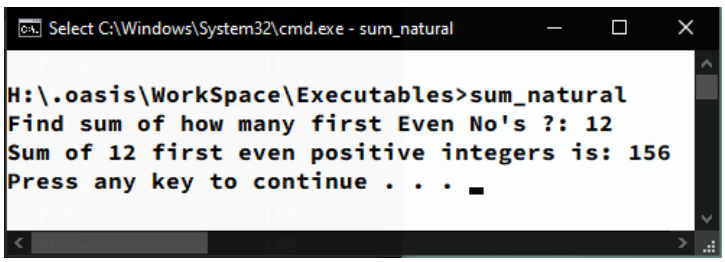
Output:



/*C++ Program to find sum of N even Integers using GOTO statement.*/

```
#include<iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int num, temp = 0, res = 0;
    cout << "Find sum of how many first Even No's ?: ";
    cin >> num;
    repeat:
        temp += 2;
        res += temp;
        if(temp < num*2)
            goto repeat;
        cout << "Sum of first " << num << " even positive integers is: "
<< res << endl;
        system("pause");
        return 0;
}
```

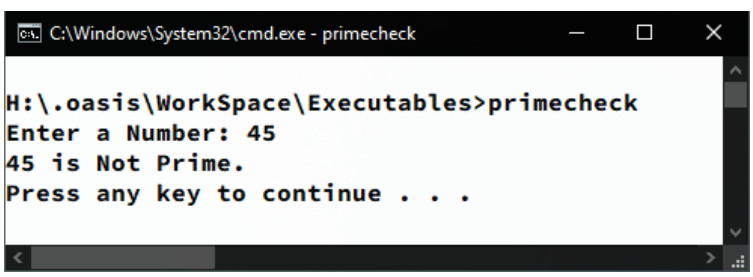
Output:



/* C++ Program to Check whether a number is Prime or Not. */

```
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int num, temp;
    bool state = true;
    cout << "Enter a Number: ";
    cin >> num;
    for (int i = 2; i <= num / 2; i++) {
        temp = num % i;
        if (temp == 0)
            state = false;
    }
    if (state) cout << num << " is Prime." << endl;
    else cout << num << " is Not Prime." << endl;
    system("pause");
    return 0;
}
```

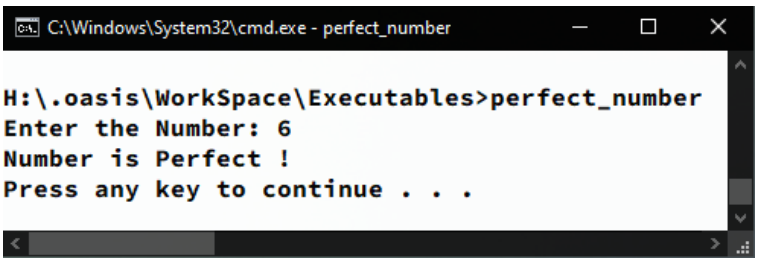
Output:

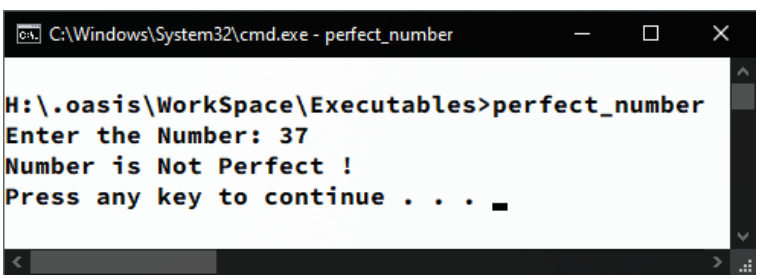


/* C++ Program to Check whether a number is Perfect or Not. */

```
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int num, temp, check = 0;
    cout << "Enter the Number: ";
    cin >> num;
    for (int i = 1; i <= num / 2; i++) {
        temp = num % i;
        if (temp == 0)
            check += i;
    }
    if (num == check)
        cout << "Number is Perfect !" << endl;
    else
        cout << "Number is Not Perfect !" << endl;
    system("pause");
    return 0;
}
```

Output:

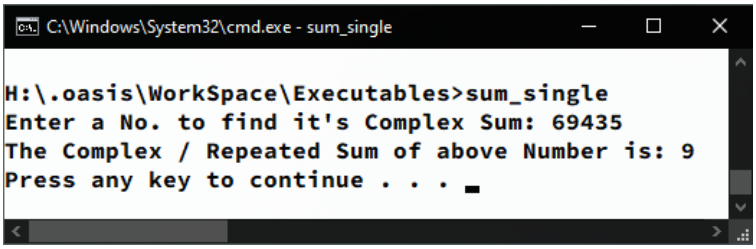




/* C++ Program to Find Complex/ Repetitive sum of a Number. */

```
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int num, rem, sum;
    cout << "Enter a No. to find it's Complex Sum: ";
    cin >> num;
    while (num > 9) {
        sum = num;
        num = 0;
        while (sum > 0) {
            rem = sum % 10;
            sum /= 10;
            num += rem;
        }
    }
    cout << "The Complex / Repeated Sum of above Number is: " << num << endl;
    system("pause");
    return 0;
}
```

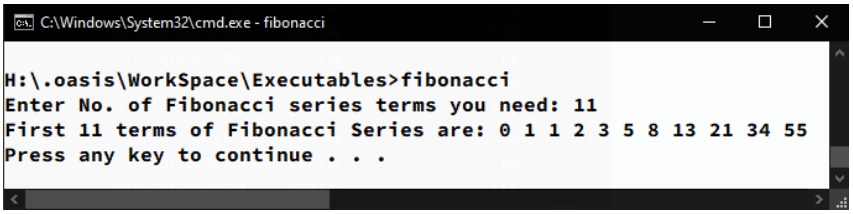
Output:



/* C++ Program to Generate N terms of Fibonacci Sequence. */

```
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int limit, current = 0, next = 1, swap;
    cout << "Enter No. of Fibonacci series terms you need: ";
    cin >> limit;
    cout << "First " << limit << " terms of Fibonacci Series are:";
    for (int i = 0; i < limit; i++) {
        cout << " " << current;
        swap = current + next;
        current = next;
        next = swap;
    }
    cout << endl;
    system("pause");
    return 0;
}
```

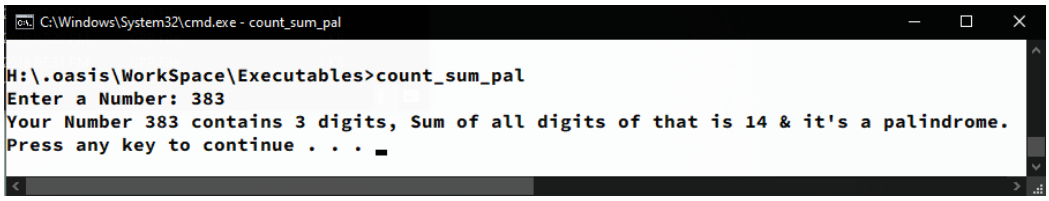
Output:



/* C++ Program to count number of digits, sum of digits & Palindrome check. */

```
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int num, rem, sum = 0, rev = 0, count = 0, temp;
    cout << "Enter a Number: ";
    cin >> num;
    temp = num;
    while (num > 0) {
        count++;
        rem = num % 10;
        sum += rem;
        rev = rev * 10 + rem;
        num /= 10;
    }
    cout << "Your Number " << temp << " contains " << count << " digits,
Sum of all digits of that is " << sum;
    if (rev == temp) cout << " & it's a palindrome";
    cout << "." << endl;
    system("pause");
    return 0;
}
```

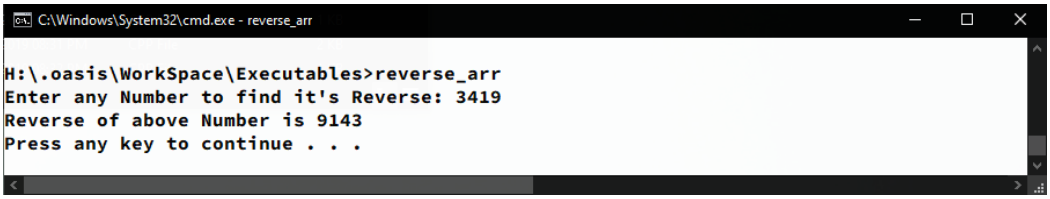
Output:



/* C++ Program to generate Reverse of a Number using Array. */

```
#include <iostream>
using namespace std;
int main(int argc, char const *argv[] {
    int num, arr[11], i = 0, j = 0;
    cout << "Enter any Number to find it's Reverse: ";
    cin >> num;
    while (num > 0) {
        arr[i] = num % 10;
        num /= 10;
        i++;
    }
    cout << "Reverse of above Number is ";
    while (j < i) {
        cout << arr[j];
        j++;
    }
    cout << endl;
    system("pause");
    return 0;
}
```

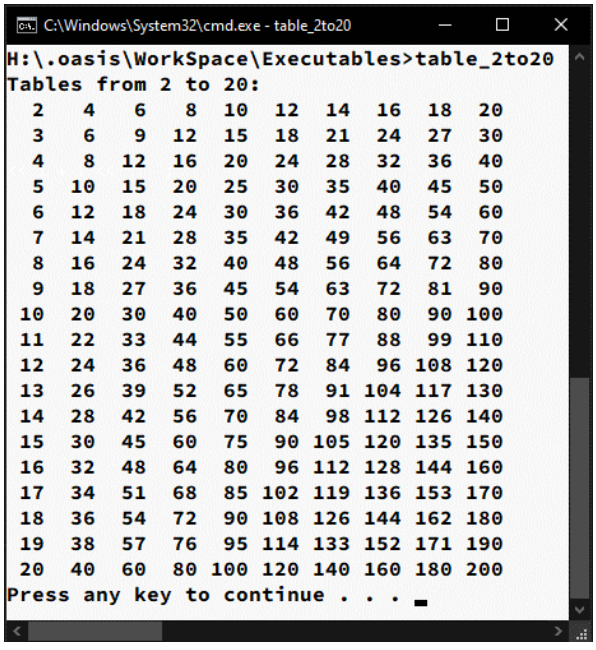
Output:



/* C++ Program to Print Multiplication table from 2 to 20. */

```
#include<iostream>
#include<iomanip>
using namespace std;
int main(int argc, char const *argv[]) {
    cout << "Tables from 2 to 20: " << endl;
    for (int i = 2; i <= 20; i++) {
        for (int j = 1; j <= 10; j++) {
            cout << setw(3) << i * j << endl;
        }
      cout << endl;
    }
    system("pause");
    return 0;
}
```

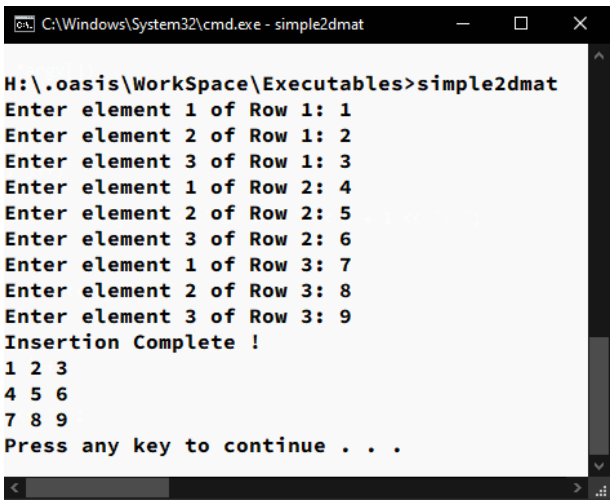
Output:



/* C++ Program to Create & Insert data into a 2D Matrix. */

```
#include<iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int mat[3][3];
    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++) {
            cout << "Enter element " << j+1 << " of Row " << i+1 << ": ";
            cin >> mat[i][j];
        }
    cout << "Insertion Complete !" << endl;
    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            cout << mat[i][j] << " ";
        cout << endl;
    system("pause");
    return 0;
}
```

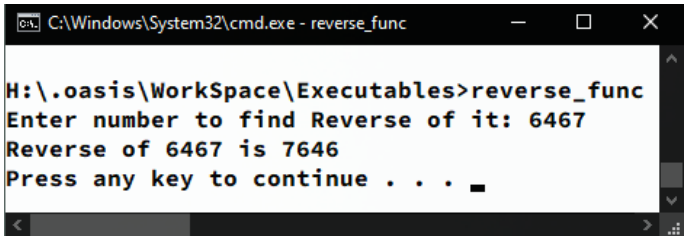
Output:



/* C++ Program to get Reverse of a Number using Function. */

```
#include<iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int num, rev(int);
    cout << "Enter number to find Reverse of it: ";
    cin >> num;
    cout << "Reverse of " << num << " is " << rev(num) << endl;
    system("pause");
    return 0;
}
int rev(int x){
    int rev = 0;
    while(x > 0){
        rev = (rev * 10) + (x % 10);
        x /= 10;
    }
    return rev;
}
```

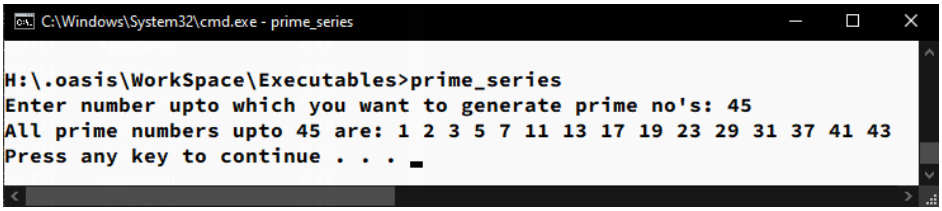
Output:



/* C++ Program to Generate Prime numbers up to N. */

```
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int max, prime;
    cout << "Enter number upto which you want to generate prime no's: ";
    cin >> max;
    cout << "All prime numbers upto " << max << " are:";
    for (int j = 1; j <= max; j++){
        prime = 1;
        for (int i = 2; i <= j/2; i++)
            if (j % i == 0) prime = 0;
        if (prime == 1)
            cout << " " << j;
    }
    cout << endl;
    system("pause");
    return 0;
}
```

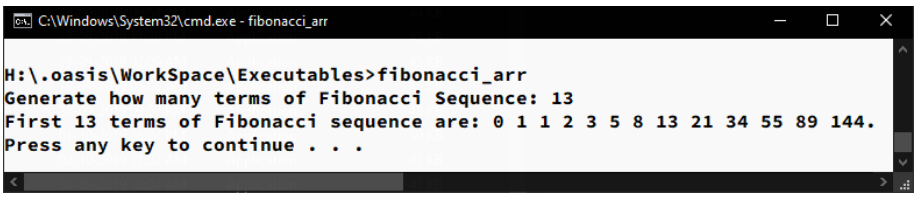
Output:



// C++ Program to Generate N terms of Fibonacci Sequence using Array.

```
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int terms, base;
    cout << "Generate how many terms of Fibonacci Sequence: ";
    cin >> terms;
    int arr[terms];
    arr[0] = 0; arr[1] = 1;
    for (int i = 1; i < terms; i++)
        arr[i + 1] = arr[i] + arr[i - 1];
    cout << "First " << terms << " terms of Fibonacci sequence are:";
    for (int i = 0; i < terms; i++)
        cout << " " << arr[i];
    cout << "." << endl;
    system("pause");
    return 0;
}
```

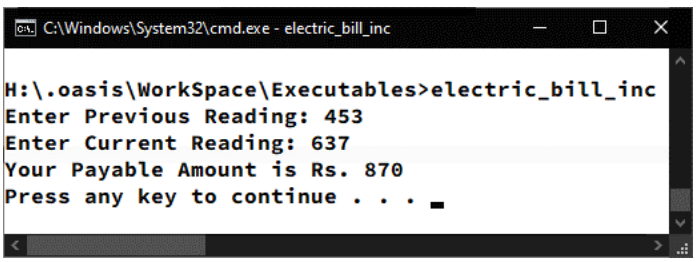
Output:



/* C++ Program to get Electric Bill amount at Variable Rates. */

```
#include<iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    float pre_read, cur_read, units, amt = 0, i = 0;
    float rate[] = {4.5, 5.0, 5.5, 6.0, 6.5, 7.5};
    cout << "Enter Previous Reading: ";
    cin >> pre_read;
    cout << "Enter Current Reading: ";
    cin >> cur_read;
    units = cur_read - pre_read;
    while (units >= 100){
        units -= 100;
        amt += 100 * rate[i++];
        if(i==6) i--;
    }
    amt += units * rate[i];
    cout << "Your Payable Amount is Rs. " << amt << endl;
    system("pause");
    return 0;
}
```

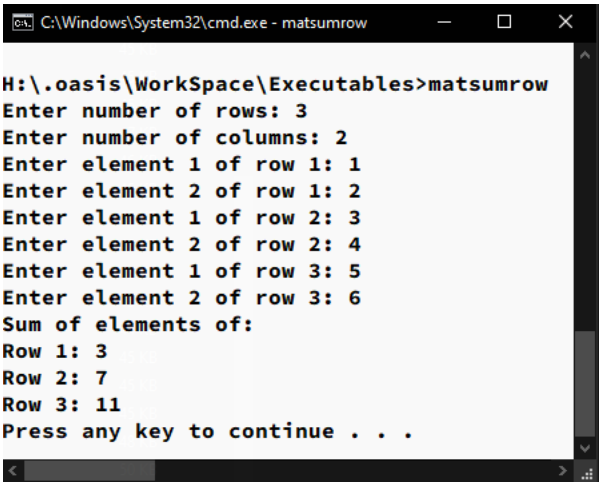
Output:



/* C++ Program to find Sum of Individual Rows of Matrix. */

```
#include<iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int rows, columns;
    cout << "Enter number of rows: ";
    cin >> rows;
    cout << "Enter number of columns: ";
    cin >> columns;
    int mat[rows][columns], sum[rows];
    for (int i = 0; i < rows; i++) {
        sum[i] = 0;
        for (int j = 0; j < columns; j++) {
        cout << "Enter element " << j + 1 << " of row " << i + 1 << ": ";
            cin >> mat[i][j]; sum[i] += mat[i][j];
        }
    }
    cout << "Sum of elements of:" << endl;
    for (int i = 0; i < rows; i++)
        cout << "Row " << i+1 << ": " << sum[i] << endl;
    system("pause");
    return 0;
}
```

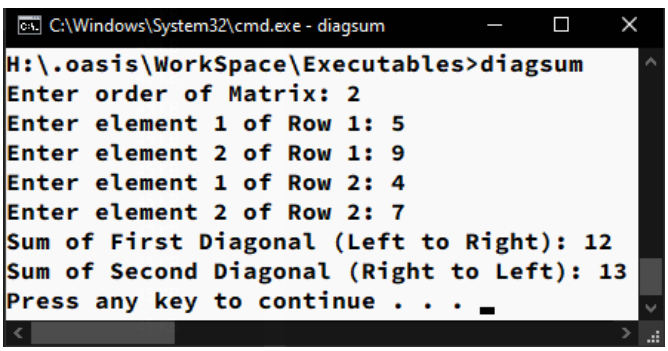
Output:



/* C++ Program to find sum of Diagonal elements of Square matrix. */

```
#include<iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int order, diag1 = 0, diag2 = 0;
    cout << "Enter order of Matrix: ";
    cin >> order; int mat[order][order];
    for (int i = 0; i < order; i++) {
        for (int j = 0; j < order; j++) {
        cout << "Enter element " << j + 1 << " of Row " << i + 1 << ": ";
            cin >> mat[i][j];
            if(i == j){
                diag1 += mat[i][j];
            } if(i+j == order-1){
                diag2 += mat[i][j];
            }
        }
    }
    cout << "Sum of First Diagonal (Left to Right): " << diag1 << endl;
    cout << "Sum of Second Diagonal (Right to Left): " << diag2 << endl;
    system("pause");
    return 0;
}
```

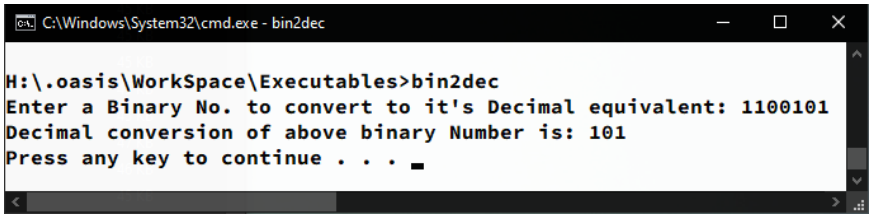
Output:



/* C++ Program to Convert Binary Number to Decimal. */

```
#include <iostream>
#include <math.h>
using namespace std;
int main(int argc, char const *argv[]) {
    int bin, dec = 0, rem, i = 0; bool halt = true;
    cout << "Enter a Binary No. to convert to it's Decimal equivalent: ";
    cin >> bin;
    while (bin > 0) {
        rem = bin % 10;
        dec += rem * pow(2, i);
        i++;
        bin /= 10;
        if (rem != 0 && rem != 1) {
            cout << "The Number you have Entered is NON-BINARY !!!" <<
endl;
            halt = false;
            break;
        }
    } if (halt)
        cout << "Decimal conversion of above binary Number is: " << dec
<< endl;
    system("pause");
    return 0;
}
```

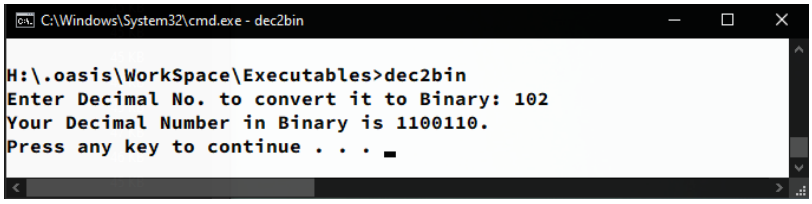
Output:



/* C++ Program to convert Decimal number to Binary. */

```
#include <iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int dec, bin[32], i = 0;
    cout << "Enter Decimal No. to convert it to Binary: ";
    cin >> dec;
    while (dec > 0) {
        bin[i] = dec % 2;
        dec /= 2;
        i++;
    }
    cout << "Your Decimal Number in Binary is ";
    for (int j = i - 1; j >= 0; j--) {
        cout << bin[j];
    }
    cout << "." << endl;
    system("pause");
    return 0;
}
```

Output:

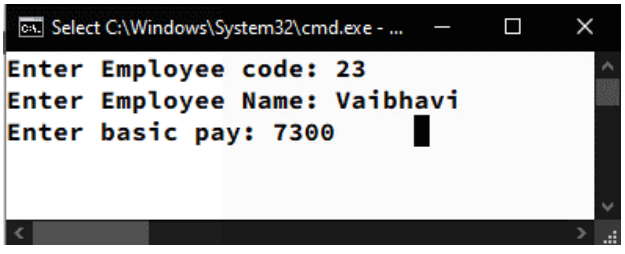


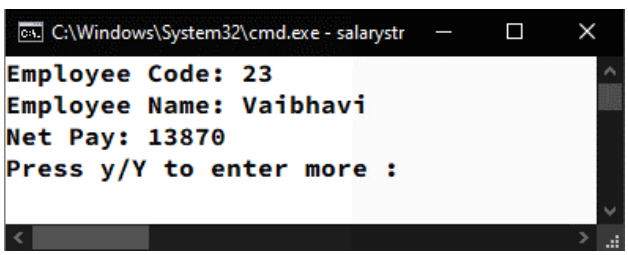
/* C++ Program to calculate Net Pay of Employee using Structure. */

```
#include<iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    struct employee{
        string name;
        int code;
        float bp;
    };
    float da, hra, net;
    char ch;
    employee emp;
    do{
        system("cls");
        cout << "Enter Employee code: ";
        cin >> emp.code;
        cout << "Enter Employee Name: ";
        cin >> emp.name;
        cout << "Enter basic pay: ";
        cin >> emp.bp;
        da = emp.bp * 0.72;
        hra = emp.bp * 0.18;
        net = emp.bp + da + hra;
        system("cls");
        cout << "Employee Code: " << emp.code << endl;
        cout << "Employee Name: " << emp.name << endl;
        cout << "Net Pay: " << net << endl;
        cout << "Press y/Y to enter more : ";
        cin >> ch;
```

```
    } while (ch == 'Y' || ch == 'y');
    system("pause");
    return 0;
}
```

Output:



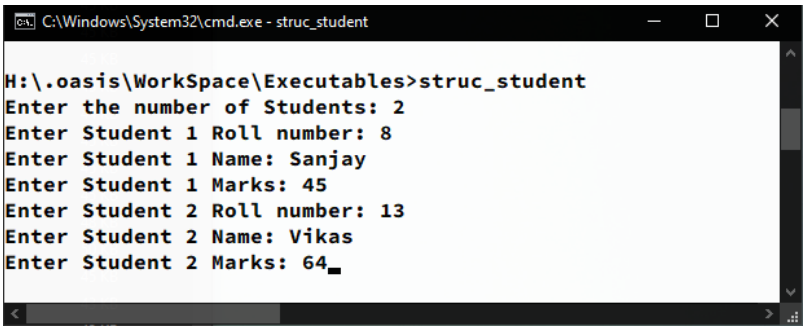


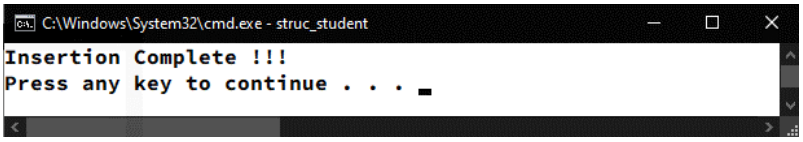
// C++ Program to Store Data of N students into an Array of Structure.

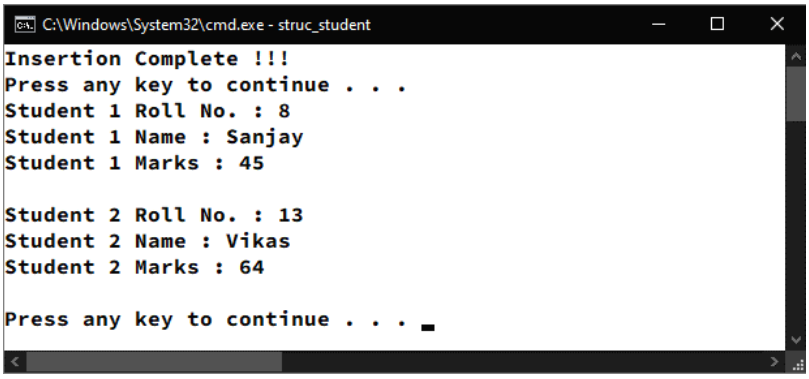
```
#include<iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    struct student{
        int roll;
        int marks;
        string name;
    };
    int num;
    cout << "Enter the number of Students: ";
    cin >> num;
    student st[num];
    for (int i = 0; i < num; i++) {
        cout << "Enter Student " << i + 1 << " Roll number: ";
        cin >> st[i].roll;
        cout << "Enter Student " << i + 1 << " Name: ";
        cin >> st[i].name;
        cout << "Enter Student " << i + 1 << " Marks: ";
        cin >> st[i].marks;
    }
    system("cls");
    cout<< "Insertion Complete !!!" << endl;
    system("pause");
    for (int i = 0; i < num; i++) {
        cout << "Student " << i + 1 << " Roll No. : " << st[i].roll <<
endl;
        cout << "Student " << i + 1 << " Name : " << st[i].name << endl;
        cout << "Student " << i + 1 << " Marks : " << st[i].marks << endl
<< endl;
```

```
    }
    system("pause");
    return 0;
}
```

Output:



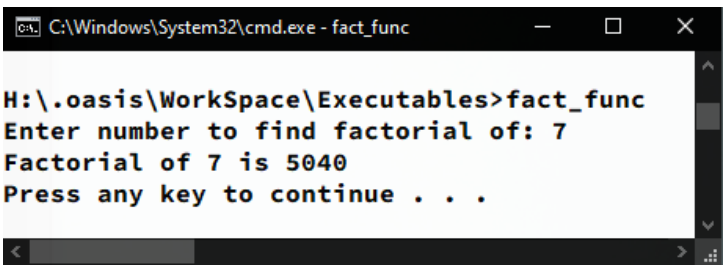




/* C++ Program to find factorial of a Number using Function. */

```
#include<iostream>
using namespace std;
int main(int argc, char const *argv[]) {
    int num, fact(int);
    cout << "Enter number to find factorial of: ";
    cin >> num;
    cout << "Factorial of " << num << " is " << fact(num) << endl;
    system("pause");
    return 0;
}
int fact(int x){
    int fact = 1;
    for (int i = 1; i <= x; i++) fact *= i;
    return fact;
}
```

Output:



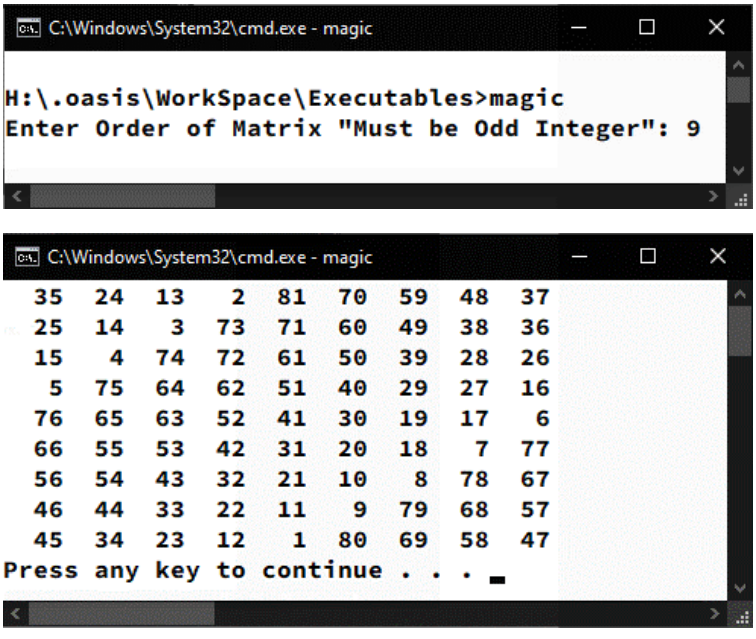
/* C++ Program to Generate & print Magic Matrix. */

```
#include<iostream>
#include<iomanip>
using namespace std;
int main(int argc, char const *argv[]) {
    int mat[15][15], order, count=1;
    bool err, end;
    do{
        if(argc==2){
            order = atoi(argv[1]);
            if(order%2 != 1){
                err = true;
                goto step;
            }
        } else {
            step:
            if(err){
                system("cls");
                cout << "Wrong Parameters !!!" << endl;
            }
               err = true;
               cout << "Enter Order of Matrix \"Must be Odd Integer\": ";
               cin >> order;
        }
    } while ((order % 2) != 1);
    for (int i = 0; i < order; i++)
        for (int j = 0; j < order; j++)
            mat[i][j] = 0;
    system("cls");
```

```
int max = order * order;
int size = order - 1;
int i = size;
int j = size / 2;

for ( ; i < order; ){
    for ( ; j < order; ){
        mat[i][j] = count;
        if(count == max){
            end = true;
            break;
        }else count++;
        i++;
        if(i > size)
            i = 0;
        j--;
        if(j < 0)
            j = size;
        if (mat[i][j] != 0){
            i--;
        if(i < 0)
            i = size;
            i--;
            j++;
        if(j > size)
            j = 0;
        }
    }
    if (end) break;
}
```

```
    for (int i = 0; i < order; i++)
        for (int j = 0; j < order; j++)
            cout << setw(4) << mat[i][j];
        cout << endl;
    system("pause");
    return 0;
}
```

Output:

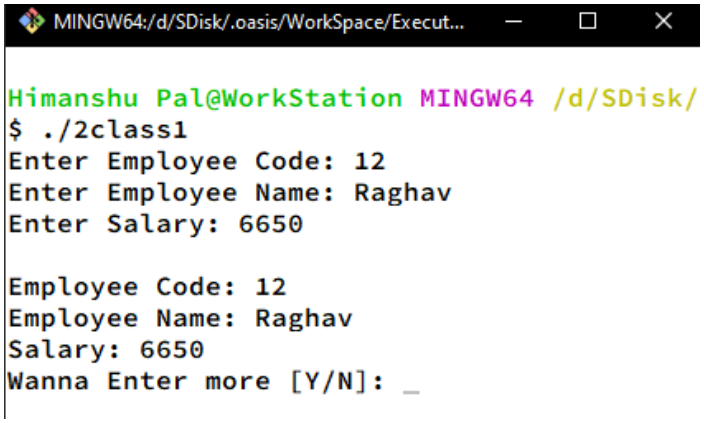


/* Program to demonstrate working of Class. */

```
#include <iostream>
using namespace std;
class employee {
private:
    int empCode;
    string name;
    float salary;
public:
    void getdata() {
        cout << "Enter Employee Code: ";
        cin >> empCode;
        cout << "Enter Employee Name: ";
        cin >> name;
        cout << "Enter Salary: ";
        cin >> salary;
    }
     void showdata() {
        system("cls");
        cout << "Employee Code: " << empCode << endl;
        cout << "Employee Name: " << name << endl;
        cout << "Salary: " << salary << endl;
    }
};

int main(int argc, char const *argv[]) {
    employee emp;
    char ch;
    do {
```

```
        emp.getdata();
        emp.showdata();
        cout << "Wanna Enter more [Y/N]: ";
        cin >> ch;
    } while (ch != 'n' && ch != 'N');
    return 0;
}
```

Output:

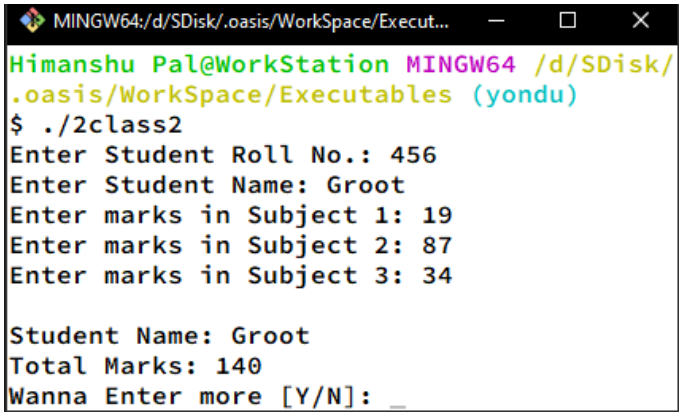


/* Program to demonstrate working of Class with functions. */

```
#include <iostream>
using namespace std;
class student {
private:
    int rollNo, mark1, mark2, mark3; string name;
public:
    void getdata();
    int calculate();
    void showdata();
};
void student::getdata() {
    cout << "Enter Student Roll No.: ";
    cin >> rollNo;
    cout << "Enter Student Name: ";
    cin >> name;
    cout << "Enter marks in Subject 1: ";
    cin >> mark1;
    cout << "Enter marks in Subject 2: ";
    cin >> mark2;
    cout << "Enter marks in Subject 3: ";
    cin >> mark3;
}
int student::calculate() {
    return mark1 + mark2 + mark3;
}
void student::showdata() {
    system("cls");
    cout << "Student Name: " << name << endl;
```

```
    cout << "Total Marks: " << calculate() << endl;
}
int main(int argc, char const *argv[]) {
    student stu;
    char ch;
    do {
        stu.getdata();
        stu.showdata();
        cout << "Wanna Enter more [Y/N]: ";
        cin >> ch;
    } while (ch != 'n' && ch != 'N');
    return 0;
}
```

Output:

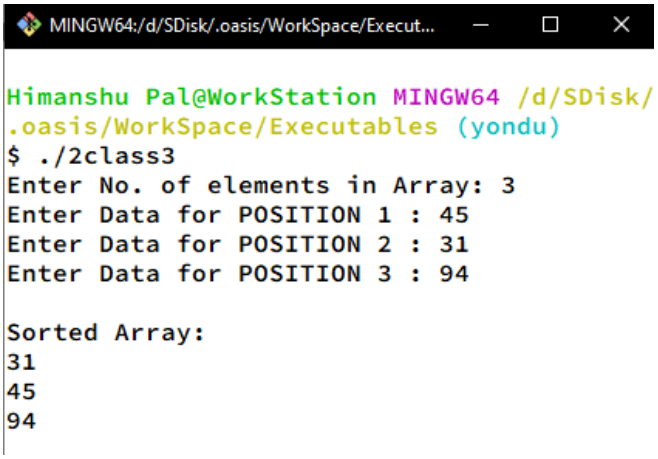


/* Program to demonstrate working of Class by Sorting an Array. */

```
#include <iostream>
using namespace std;
int limit;
class sort {
private:
    int pos = 0, arr[100];
public:
    void getdata();
    void calc();
    void showdata();
};
void sort::getdata() {
    while (pos < limit) {
        cout << "Enter Data for POSITION " << pos+1 << " : ";
        cin >> arr[pos];
        pos -= -1;
    } }
void sort::calc() {
    for (int i = 0; i < limit - 1; i++) {
        for (int j = i; j < limit; j++) {
            if (arr[i] > arr[j]) {
                arr[i] += arr[j];
                arr[j] = arr[i] - arr[j];
                arr[i] -= arr[j];
            } } }
     }
void sort::showdata() {
    system("cls");
```

```
    cout << "Sorted Array:" << endl;
    for (int i = 0; i < limit; i++) {
        cout << arr[i] << endl;
    }
}
int main(int argc, char const *argv[]) {
    sort srt;
    cout << "Enter No. of elements in Array: ";
    cin >> limit;
    srt.getdata();
    srt.calc();
    srt.showdata();
    return 0;
}
```

Output:

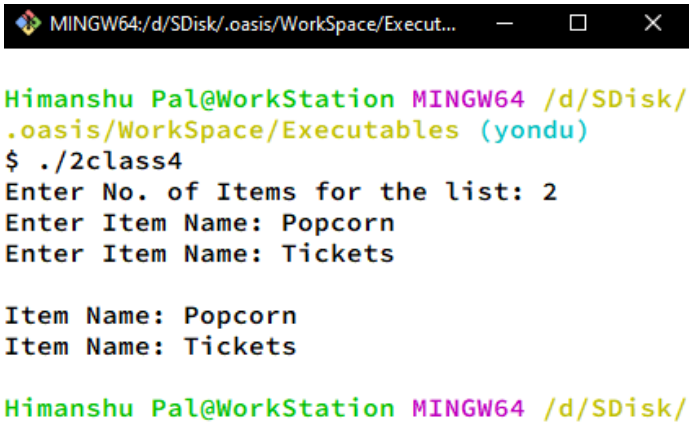


/* Program to demonstrate working of Class by creating an Array of Objects. */

```
#include <iostream>
using namespace std;

class list
{
private:
    string data;

public:
    void getdata();
    void showdata();
};

void list::getdata()
{
    cout << "Enter Item Name: ";
    cin >> data;
}

void list::showdata()
{
    cout << "Item Name: " << data << endl;
}

int main(int argc, char const *argv[])
{
    int qty;
```

```
    cout << "Enter No. of Items for the list: ";
    cin >> qty;
    list item[qty];
    for (int i = 0; i < qty; i++)
    {
        item[i].getdata();
    }
    system("cls");
    for (int i = 0; i < qty; i++)
    {
        item[i].showdata();
    }
    return 0;
}
```

Output:

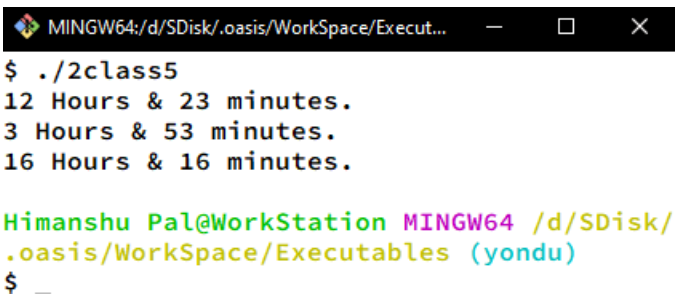


/* Program to demonstrate working of Class passing object as an argument. */

```
#include <iostream>
using namespace std;

class time
{
private:
    int hour, minute;

public:
    void get_time(int i, int j)
    {
        hour = i;
        minute = j;
    }
    void showtime()
    {
        cout << hour << " Hours & " << minute << " minutes." << endl;
    }
    void sum(time t1, time t2)
    {
        minute = t1.minute + t2.minute;
        hour = minute / 60;
        minute %= 60;
        hour = hour + t1.hour + t2.hour;
    }
};
```

```
int main(int argc, char const *argv[])
{
    time t1, t2, t3;
    t1.get_time(12, 23);
    t1.showtime();
    t2.get_time(3, 53);
    t2.showtime();
    t3.sum(t1, t2);
    t3.showtime();
    return 0;
}
```

Output:

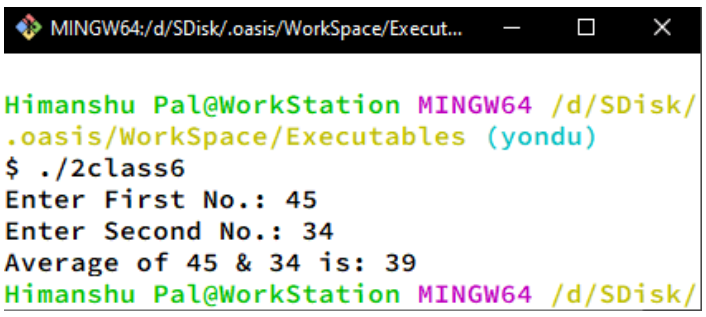


/* Program to demonstrate working of Class with friend function. */

```
#include <iostream>
using namespace std;

class test
{
private:
    int num1, num2;

public:
    void get_data(int i, int j)
    {
        num1 = i;
        num2 = j;
    }
    friend int mean(test);
};

int mean(test obj)
{
    return (obj.num1 + obj.num2) / 2;
}

int main(int argc, char const *argv[])
{
    test ob;
    int num1, num2;

    cout << "Enter First No.: ";
```

```
    cin >> num1;
    cout << "Enter Second No.: ";
    cin >> num2;
    ob.get_data(num1, num2);
    cout << "Average of " << num1 << " & " << num2 << " is: " <<
mean(ob);
    return 0;
}
```

Output:

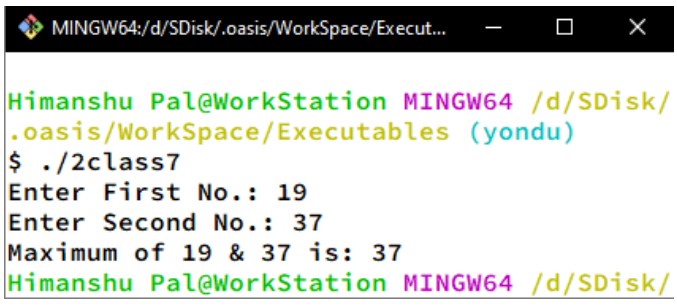


/* Program to demonstrate working of Class with friend function working as bridge. */

```
#include <iostream>
using namespace std;
class test1;
class test2;
class test1
{
private:
    int num;
public:
    void set_val(int i) {
        num = i;
    }
    friend int max(test1, test2);
};
class test2 {
private:
    int num;
public:
    void set_val(int i) {
        num = i;
    }
    friend int max(test1, test2);
};
int max(test1 obj1, test2 obj2) {
    return ((obj1.num > obj2.num) ? obj1.num : obj2.num);
}
int main(int argc, char const *argv[]) {
```

```
    test1 ob1;
    test2 ob2;
    int num1, num2;

    cout << "Enter First No.: ";
    cin >> num1;
    cout << "Enter Second No.: ";
    cin >> num2;
    ob1.set_val(num1);
    ob2.set_val(num2);
    cout << "Maximum of " << num1 << " & " << num2 << " is: " << max(ob1,
ob2);
    return 0;
}
```

Output:

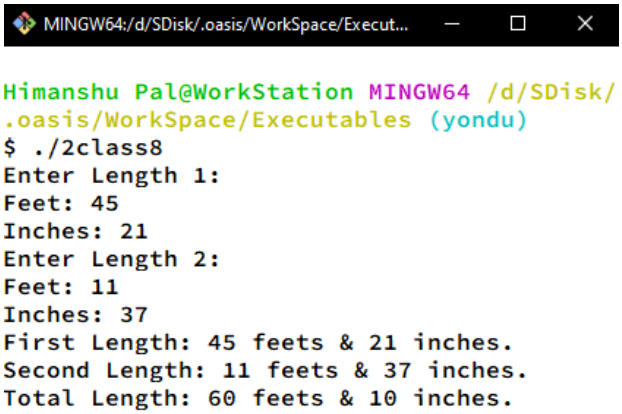


// Program to demonstrate working of Class returning object as Result.

```
#include <iostream>
using namespace std;

class length {
private:
    int feet; float inch;
public:
    void get_data(int ft, float in) {
        feet = ft;
        inch = in;
    }
    void show_data() {
        cout << feet << " feets & " << inch << " inches." << endl;
    }
    length total(length l1, length l2);
};
length length::total(length l1, length l2) {
    length l3;
    l3.feet = l1.feet + l2.feet;
    l3.inch = l1.inch + l2.inch;
    while (l3.inch > 12)
        if (l3.inch >= 12) {
            l3.feet++;
            l3.inch -= 12;
        }
    return l3;
}
int main(int argc, char const *argv[]) {
```

```
    length len1, len2; int feet; float inch;
    cout << "Enter Length 1: " << endl;
    cout << "Feet: "; cin >> feet;
    cout << "Inches: "; cin >> inch;
    len1.get_data(feet, inch);
    cout << "Enter Length 2: " << endl;
    cout << "Feet: "; cin >> feet;
    cout << "Inches: "; cin >> inch;
    len2.get_data(feet, inch);
    cout << "First Length: ";
    len1.show_data();
    cout << "Second Length: ";
    len2.show_data();
    length len3 = len3.total(len1, len2);
    cout << "Total Length: ";
    len3.show_data();
    return 0;
}
```

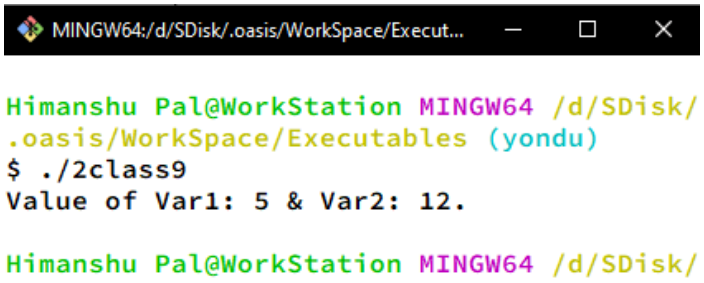
Output:



/* Program to demonstrate working of Class using Constructor. */

```
#include <iostream>
using namespace std;
class test {
private:
    int num1 = 5, num2 = 12;
public:
    test(){
        cout << "Value of Var1: " << num1 << " & Var2: " << num2 << "."
<< endl;
    }
};
int main(int argc, char const *argv) {
    test obj;
    return 0;
}
```

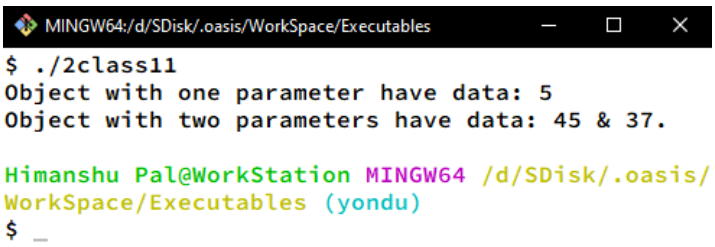
Output:



/* Program to demonstrate working of Class using multiple parameterised Constructor. */

```
#include <iostream>
using namespace std;
class sample {
private:
    int num1, num2;
public:
    sample(int a) {
        num1 = a;
        cout << "Object with one parameter have data: " << num1 << endl;
    } sample(int a, int b) {
        num1 = a;
        num2 = b;
        cout << "Object with two parameters have data: " << num1 << " & "
<< num2 << "." << endl;
    }
};
int main(int argc, char const *argv[]) {
    sample obj1(5);
    sample obj2(45, 37);
    return 0;
}
```

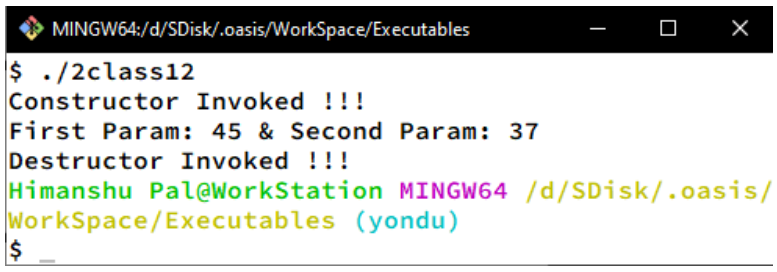
Output:



/* Program to demonstrate working of Class using Constructor & Destructor. */

```
#include <iostream>
using namespace std;
class sample {
private:
    int num1, num2;
public:
    sample(int a, int b) {
        num1 = a;
        num2 = b;
        cout << "Constructor Invoked !!!\n";
        cout << "First Param: " << num1 << " & Second Param: " << num2 <<
endl;
    }
    ~sample(){
        cout << "Destructor Invoked !!!";
    }
};
int main(int argc, char const *argv[]) {
    sample obj2(45, 37);
    return 0;
}
```

Output:

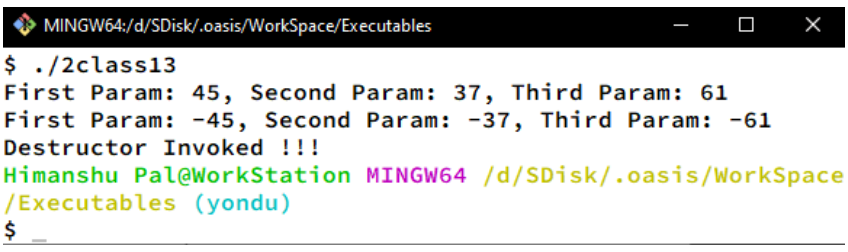


// Program to demonstrate Operator Overloading for Unary Operator.

```
#include <iostream>
using namespace std;

class sample
{
private:
    int num1, num2, num3;

public:
    sample(int a, int b, int c)
    {
        num1 = a;
        num2 = b;
        num3 = c;
    }
    void display()
    {
        cout << "First Param: " << num1 << ", Second Param: " << num2 <<
", Third Param: " << num3 << endl;
    }
    void operator-()
    {
        num1 = -num1;
        num2 = -num2;
        num3 = -num3;
    }
    ~sample()
    {
```

```
        cout << "Destructor Invoked !!!";
    }
};

int main(int argc, char const *argv[])
{
    sample obj(45, 37, 61);
    obj.display();
    -obj;
    obj.display();
    return 0;
}
```

Output:

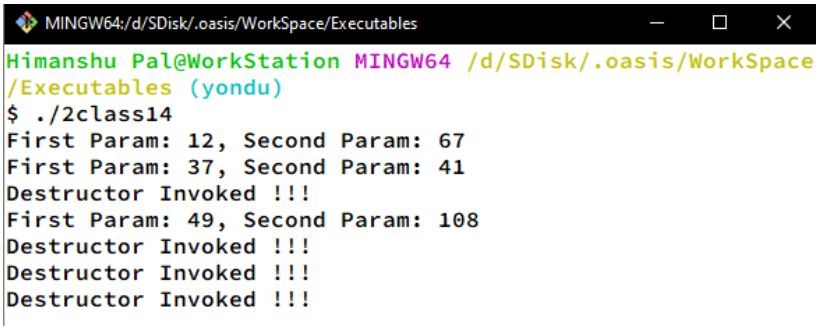


// Program to demonstrate Operator Overloading for Binary Operator.

```
#include <iostream>
using namespace std;

class sample
{
private:
    int num1, num2;

public:
    sample(int a, int b)
    {
        num1 = a;
        num2 = b;
    }
    void display()
    {
        cout << "First Param: " << num1 << ", Second Param: " << num2 <<
endl;
    }
    sample operator+(sample ob)
    {
        sample obj(0, 0);
        obj.num1 = num1 + ob.num1;
        obj.num2 = num2 + ob.num2;
        return obj;
    }
    ~sample()
    {
```

```
        cout << "Destructor Invoked !!!\n";
    }
};

int main(int argc, char const *argv[])
{
    sample obj1(12, 67), obj2(37, 41);
    obj1.display();
    obj2.display();
    sample obj = obj1 + obj2;
    obj.display();
    return 0;
}
```

Output:



// Program to demonstrate Operator Overloading for Relational Operator.

```
#include <iostream>
using namespace std;

class currency
{
private:
    int rs, ps;

public:
    currency(int a, int b)
    {
        rs = a;
        ps = b;
    }
    void display()
    {
        cout << rs << " rupees & " << ps << " paise." << endl;
    }
    bool operator>(currency ob)
    {
        rs += ps / 100;
        ob.rs += ob.ps / 100;
        return ((rs > ob.rs) ? true : false);
    }
    ~currency()
    {
        cout << "Destructor Invoked !!!" << endl;
```

```
    }
};

int main(int argc, char const *argv[])
{
    currency obj1(13, 67), obj2(37, 141);
    obj1.display();
    obj2.display();
    if (obj1 > obj2)
        cout << "First Amt. is larger." << endl;
    else
        cout << "Second Amt is larger." << endl;
    return 0;
}
```

Output: