



غلام ایشاق خان انسٹیٹیوٹ آف انجینئرنگ سائنس و ٹیکنالوجی  
Ghulam ishaq khan institute of engineering sciences and technology

**NAME:** MUHAMMAD HARIS

**CLASS:** CS101

**COURSE:** 2023428

---

### Q No 01(CLO # 2, PLO #1)

Write the Program that prints the number in reverse order. For example, your program asks the user for input i.e., 7654, and then the program should output 4567. **Use** Proper logic instead of “cout”.

---

### **CODE:**

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

int main()
{
    int num;
    int reversenum=0;
    cout<<"WELCOME TO THE OPPOSITE DIMENSION :)"<<endl;
    cout<<"ENTER THE NUMBER WANTED TO BE REVERSED"<<endl;
    cin>>num;

    while(num>0)
    {
        int digit = num%10;
        reversenum=reversenum*10+digit;
        num/=10;
    }
    cout<<reversenum<<endl;
}
```

### **EXPLANATION:**

In this code we solved the problem by initially initializing an integer that was named as 'num' and would store the value taken from the user that is to be reversed. Moreover, we have initialized the reversed num with '0'. After that we take the value from the user and save it in the int num,

Now. At the major part of the code, we use the while loop to execute our problem giving the initial condition expression to the code as  $\text{num} > 0$ , obviously a positive number can be reversed.

Then, we initialize another term named 'digit' which is equal to the modulus of the user entered number with 10, this would give us the last term of the number entered by the user as the remainder and now equal to the digit.

The code will take this first digit and multiply initially with 0 the first value of reverse number  $\times 10$  and add it to the digit, resulting in the last digit to be the first of the user entered number. And, so on.

Briefly the overview of the while loop iterations is given below with an example:

1. First Iteration:

- $\text{digit} = 7654 \% 10$  gives  $\text{digit} = 4$ .
- $\text{reversedNum} = 0 * 10 + 4$  gives  $\text{reversedNum} = 4$ .
- $\text{num} /= 10$  gives  $\text{num} = 765$ .

2. Second Iteration:

- $\text{digit} = 765 \% 10$  gives  $\text{digit} = 5$ .
- $\text{reversedNum} = 4 * 10 + 5$  gives  $\text{reversedNum} = 45$ .
- $\text{num} /= 10$  gives  $\text{num} = 76$ .

3. Third Iteration:

- $\text{digit} = 76 \% 10$  gives  $\text{digit} = 6$ .
- $\text{reversedNum} = 45 * 10 + 6$  gives  $\text{reversedNum} = 456$ .
- $\text{num} /= 10$  gives  $\text{num} = 7$ .

4. Fourth Iteration:

- `digit = 7 % 10` gives `digit = 7`.
- `reversedNum = 456 * 10 + 7` gives `reversedNum = 4567`.
- `num /= 10` gives `num = 0`.

Now, the loop stops because `num` is no longer greater than 0. The original number (7654) has been reversed, and the reversed number (4567) is the result.

```
321

d:\CSS\Fancy Button - CodeAlong>cd "d:\CSS\Fancy Button - CodeAlong\" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "d:\CSS\Fancy Button - CodeAlong\tempCodeRunnerFile
WELCOME TO THE OPPOSITE DIMENSION :)
ENTER THE NUMBER WANTED TO BE REVERSED
7654
4567

d:\CSS\Fancy Button - CodeAlong>
```

**Q No 02 (CLO # 2, PLO #1)**

Write a program that prints the following patterns separately one below the other. **Use nested** for loops to generate the patterns.

Figure 1 consists of two panels, (A) and (B), each showing a distribution of star ratings (0 to 10) for 10 categories. In panel (A), the number of stars increases from 1 to 10 for categories 1 to 10. In panel (B), the number of stars decreases from 10 to 1 for categories 1 to 10.

**A. CODE:**

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

int main()
{
```

```

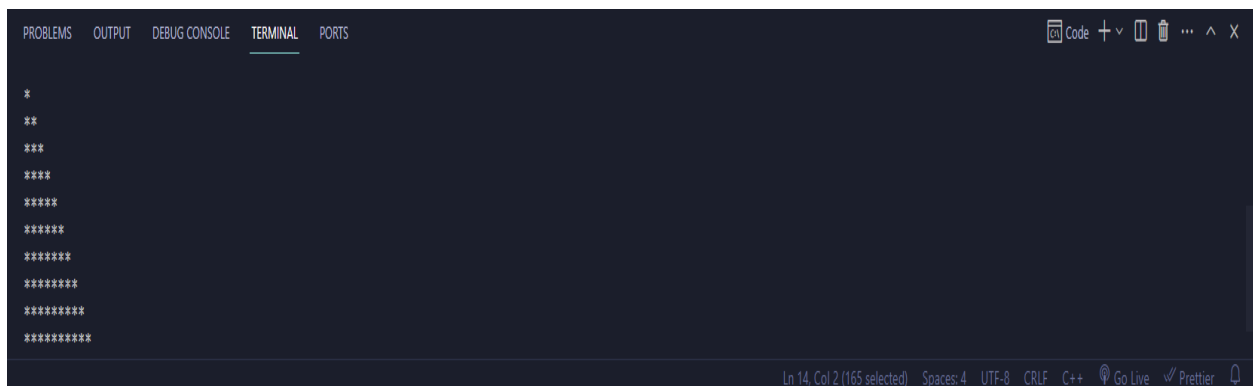
    for(int i=1; i<=10; i++)
    {
        for(int j=1; j<=i; j++)
        {
            cout<<"*";
        }
        cout<<endl;
    }
}

```

### EXPLANATION:

In this code we have used the nested for loops to make the required pattern of stars. It is to be noted that 'i' represented by rows and 'j' represented by the columns thus as the question suggests the j will be less than i at each of the interval to print the \* pattern correctly.

The program requires 10 rows with the no. of stars increasing at each row.



```

*
**
***
****
*****
*****
*****
*****
*****
*****

```

### B. CODE:

```

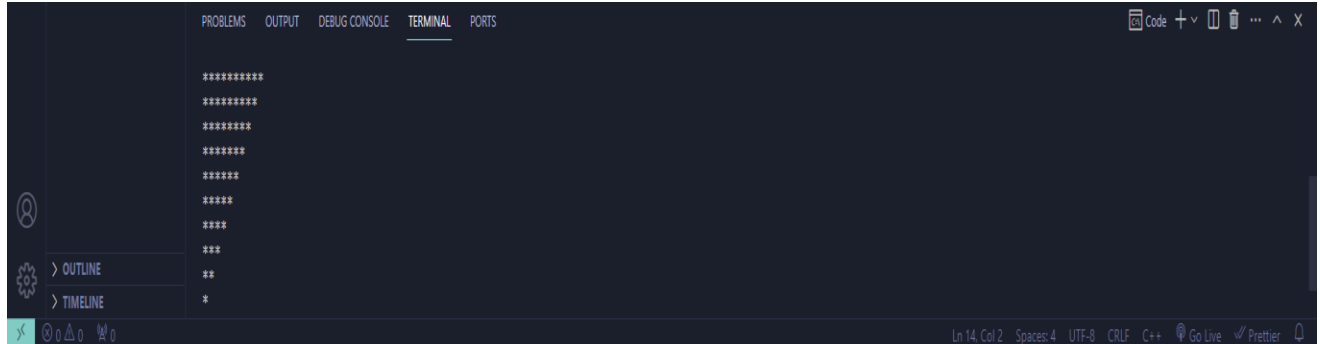
int main()
{
    for(int i=1; i<=10; i++)
    {
        for(int j=10; i<=j; j--)
        {
            cout<<"*";
        }
        cout<<endl;
    }
}

```

### EXPLANATION

In this code we have used the nested for loops to make the required pattern of stars. It is to be noted that 'i' represented by rows and 'j' represented by the columns thus as the question suggests the j will be less than i at each of the interval to print the \* pattern correctly.

The program requires 10 rows with the no. of stars decreasing at each row.

A screenshot of a code editor interface. The editor shows a pattern of stars in a 10x10 grid. The first row has 10 stars, the second row has 9 stars, and so on, down to the tenth row which has 1 star. The editor has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing the star pattern. The editor also has a sidebar with icons for a user, settings, and a list of files (OUTLINE, TIMELINE). The status bar at the bottom shows 'Ln 14, Col 2', 'Spaces: 4', 'UTF-8', 'CRLF', 'C++', 'Go Live', 'Prettier', and a refresh icon.

---

### Q No 03 (CLO # 2,PLO #1)

Write a program that prints a table of the binary, octal and hexadecimal equivalents of the decimal numbers in the range 1 through 256.

---

### CODE:

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

int main()
{
    cout<<"Table:\n";
    cout<<left<<setw(20)<<"Decimal:"<<setw(20)<<"Binary:"<<setw(20)<
    <<"Octal:"<< setw(20) << "Hexadecimal:" << endl;

    for (int i = 1; i <= 256; i++)
    {

        int sumB = 0, localBinary = 1;
        int OctaNum = i, BinaryNum = i, HexaNum = i;
```

```

//CONVERTING to Binary
while (BinaryNum!= 0)
{
    int binary = BinaryNum % 2;
    sumB = sumB + binary * localBinary;
    //REVERTING
    localBinary = localBinary * 10;
    BinaryNum = BinaryNum / 2;
}

int sumOcta = 0, localOcta = 1;

//Convert to Octal
while (OctaNum != 0)
{
    int Octa = OctaNum % 8;
    sumOcta = sumOcta + Octa * localOcta;
    //REVERTING
    localOcta = localOcta * 10;
    OctaNum = OctaNum / 8;
}

string hexResult = "";

// CONVERTING to hexadecimal
while (HexaNum != 0)
{
    int result = HexaNum % 16;
    if (result < 10)
    {
        hexResult = char(result + '0') + hexResult; //
Adding the hexaDecial to string HexResult. Beginning of HexDecimal
    }
    else
    {

```

```

        hexResult = char(result - 10 + 'A') + hexResult; //
Adding the hexaDecial to string HexResult. Beginning of HexDecimal
    }
    HexaNum = HexaNum / 16;
}

    cout << left << setw(20) << i << setw(20) << sumB <<
setw(20) << sumOcta << setw(20) << hexResult << endl;
}
}

```

### **EXPLANATION:**

In this program we are required to convert decimal numbers into binary numbers first we do that by first taking mod with 2 and then we divide by 2 to find the next binary digit, however to print it in reverse/required order, we use 'sumB' to reverse it. Similar is the case for octal form in which we just change the 2 with 8 at every point.

However, for hexa we replace 8 with 16 and add a condition to check for letters the condition number mod greater than 9, in this case we use string and ASCII code representation to reverse it.

---

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
d:\CSS\Fancy Button - CodeAlong>cd "d:\CSS\Fancy Button - CodeAlong\" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "d:\CSS\Fancy Button - CodeAlong\"tempCodeRunnerFile
Table:
Decimal:      Binary:      Octal:      Hexadecimal:
1             1             1             1
2             10            2             2
3             11            3             3
4             100           4             4
5             101           5             5
6             110           6             6
7             111           7             7
8             1000          10            8
9             1001          11            9
10            1010          12            A
11            1011          13            B
12            1100          14            C
13            1101          15            D
14            1110          16            E
15            1111          17            F
16            10000         20            10
17            10001         21            11
18            10010         22            12
19            10011         23            13
20            10100         24            14
21            10101         25            15
22            10110         26            16
23            10111         27            17
24            11000         30            18
25            11001         31            19
26            11010         32            1A
27            11011         33            1B
28            11100         34            1C
29            11101         35            1D
30            11110         36            1E
31            11111         37            1F
32            100000        40            20
33            100001        41            21
34            100010        42            22
35            100011        43            23
36            100100        44            24
37            100101        45            25
38            100110        46            26
39            100111        47            27
40            101000        50            28
41            101001        51            29
42            101010        52            2A
43            101011        53            2B
44            101100        54            2C
45            101101        55            2D
46            101110        56            2E
47            101111        57            2F
48            110000        60            30
49            110001        61            31
50            110010        62            32
```

---

#### Q.No.04 (CLO # 2,PLO #1)

A mail-order house sells five different products whose retail prices are

product 1 — \$2.98,  
product 2—\$4.50,  
product 3—\$9.98,  
product 4—\$4.49  
product 5—\$6.87

Write a program that reads a series of pairs of numbers as follows:

- a) Sold Product number
- b) Quantity sold for one day

Your program should use a switch statement to help determine the retail price for each product. Your program should **calculate** and display the total retail value of all products sold last week.

---



## CODE:

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

int main()
{
    double priceprod1 = 2.98, priceprod2 = 4.50, priceprod3 = 9.98, priceprod4 = 4.49, priceprod5 = 6.87;

    int productselected;
    int quantitysold;
    double dailyRetailPrice = 0.0;
    double weeklyRetailPrice = 0.0;

    for (int day = 1; day <= 7; day++)
    {
        cout << "DAY " << day << ": FOLLOWING IS THE PRODUCT PRICE LIST. SELECT FROM 1-5 SOLD TODAY" << endl;
        cout << "1. PRODUCT-1 (VALUE:$2.98)" << endl;
        cout << "2. PRODUCT-2 (VALUE:$4.50)" << endl;
        cout << "3. PRODUCT-3 (VALUE:$9.98)" << endl;
        cout << "4. PRODUCT-4 (VALUE:$4.49)" << endl;
        cout << "5. PRODUCT-5 (VALUE:$6.87)" << endl;

        do
        {
            cout << "Enter product number (or enter 0 to end sales for the day): ";
            cin >> productselected;

            if (productselected >= 1 && productselected <= 5)
            {
                cout << "Enter the quantity of PRODUCT " << productselected << " sold: ";
                cin >> quantitysold;

                switch (productselected)
                {
                    case 1:
                        dailyRetailPrice += quantitysold * priceprod1;
                        break;
                    case 2:
                        dailyRetailPrice += quantitysold * priceprod2;
```

```

        break;
    case 3:
        dailyRetailPrice += quantitysold * priceprod3;
        break;
    case 4:
        dailyRetailPrice += quantitysold * priceprod4;
        break;
    case 5:
        dailyRetailPrice += quantitysold * priceprod5;
        break;
    }
}
else if (productselected != 0)
{
    cout << "Invalid product number. Please enter a number between 1-5 or 0 to end sales for the day." << endl;
}

} while (productselected != 0);

weeklyRetailPrice += dailyRetailPrice;

cout << "TOTAL RETAIL PRICE FOR DAY " << day << ": $" << dailyRetailPrice << endl;
dailyRetailPrice = 0.0; // Reset for the next day
}

cout << "TOTAL RETAIL PRICE FOR THE WEEK: $" << weeklyRetailPrice << endl;

return 0;
}

```

### EXPLANATION:

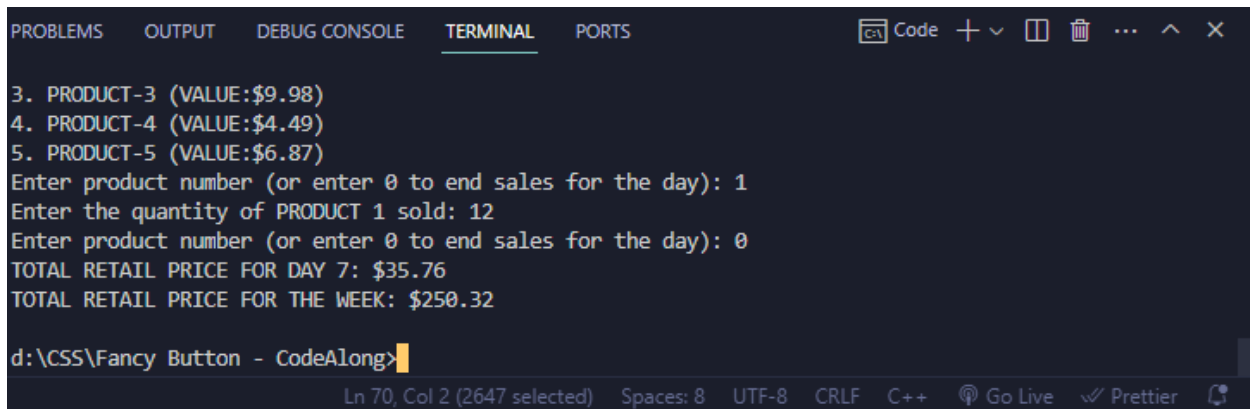
The basic idea of the program is to ask the user for which product he sold in a day and to show the sales of that day, but the twist is to repeat this act for 7 days and eventually showing the output of the different products sold in the course of 7 days, summing the revenue of each single/individual day.

We have to take care of the fact that the user can sell 2,3 or more products in a single day.

To overcome this problem, we ask the user in the do part of the do-while loop that which product he sold.

Moving on, if the user selects 1-5 as the product number, the program will again ask to tell how much of that certain product is sold by the user. Correspondingly it is to be noted that the switch statement is working to calculate the amount of revenue generated by the user on that specific product. Okay, now finally we bring the buddy of do, i.e., while in our code and we used to end the sales of that specific day only if zero is pressed by the user and the program will come out of the do while loop. Now the program being sequential will give the user revenue generated at that day. It is to be noted that the daily revenue of each day is to be displayed after the user entry of each day that is why it is to be resettled to zero every once the specific day is completed. However, we also need to find the weekly revenue generated by the user so, in order to keep all the daily revenues intact we use deposit each day's value to the weekly revenue variable and save it.

The loop runs for 7 days and eventually the weekly revenue is generated.

A screenshot of a code editor's terminal window. The terminal shows the output of a C++ program. It lists three products with their values: PRODUCT-3 (\$9.98), PRODUCT-4 (\$4.49), and PRODUCT-5 (\$6.87). It then prompts the user to enter a product number or 0 to end sales for the day. The user enters 1, and the program asks for the quantity of PRODUCT 1 sold, which is 12. The user then enters 0 to end sales for the day. The program displays the total retail price for day 7 as \$35.76 and the total retail price for the week as \$250.32. The terminal window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (selected), and PORTS. The status bar at the bottom shows 'Ln 70, Col 2 (2647 selected)', 'Spaces: 8', 'UTF-8', 'CRLF', 'C++', 'Go Live', and 'Prettier'.

---

### Q No 05 (CLO # 2, PLO #1)

**Write** a program in C++ to print a Multiplication table of any number.

**For Example, output should be something like this:**

Enter the input for the table which you want to print: 5

Multiplication table of 5

---

### CODE:

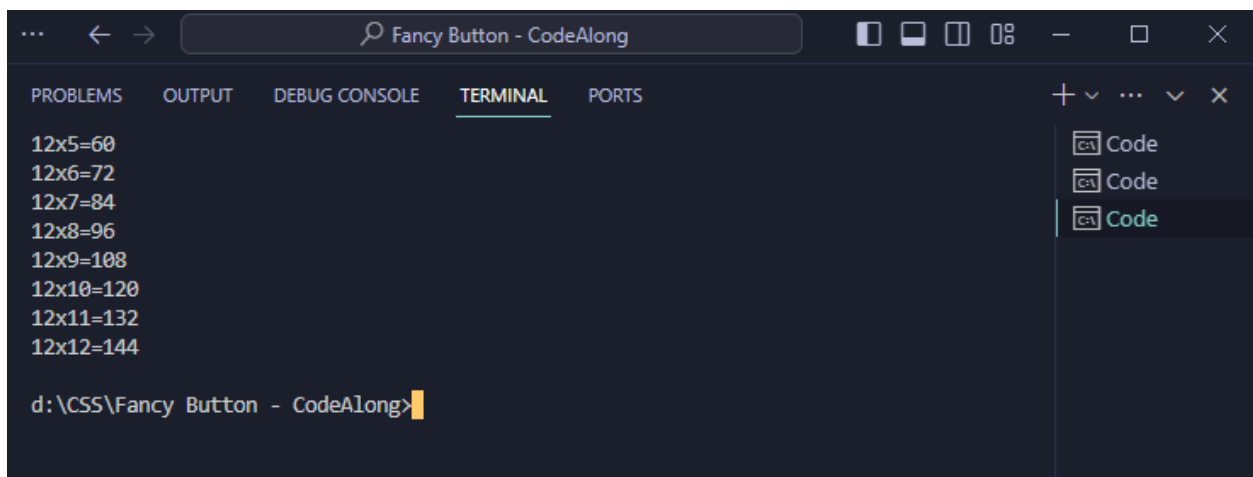
```
#include <iostream>
using namespace std;

int main()
{
    int number;
    cout<<"ENTER THE NUMBER TO FORM ITS TABLE"<<endl;
    cin>>number;

    for(int i=1; i<=12; i++)
    {
        cout<<number<<"x"<<i<<"="<<number*i<<endl;
    }
    return 0;
}
```

### EXPLANATION:

We made a multiplication table by simply using for loop running up to 12 multiples. We provide proper format to make it look like a table.

The screenshot shows a code editor window titled "Fancy Button - CodeAlong". The "TERMINAL" tab is active, displaying the output of the program. The output consists of 12 lines of multiplication results, starting from 12x5=60 and ending at 12x12=144. The prompt "d:\CSS\Fancy Button - CodeAlong>" is visible at the bottom of the terminal. On the right side of the terminal, there are three tabs labeled "Code" with a C++ icon, indicating that the source code is open in the editor.

```
12x5=60
12x6=72
12x7=84
12x8=96
12x9=108
12x10=120
12x11=132
12x12=144

d:\CSS\Fancy Button - CodeAlong>
```