Programming fundamentals

Section 4

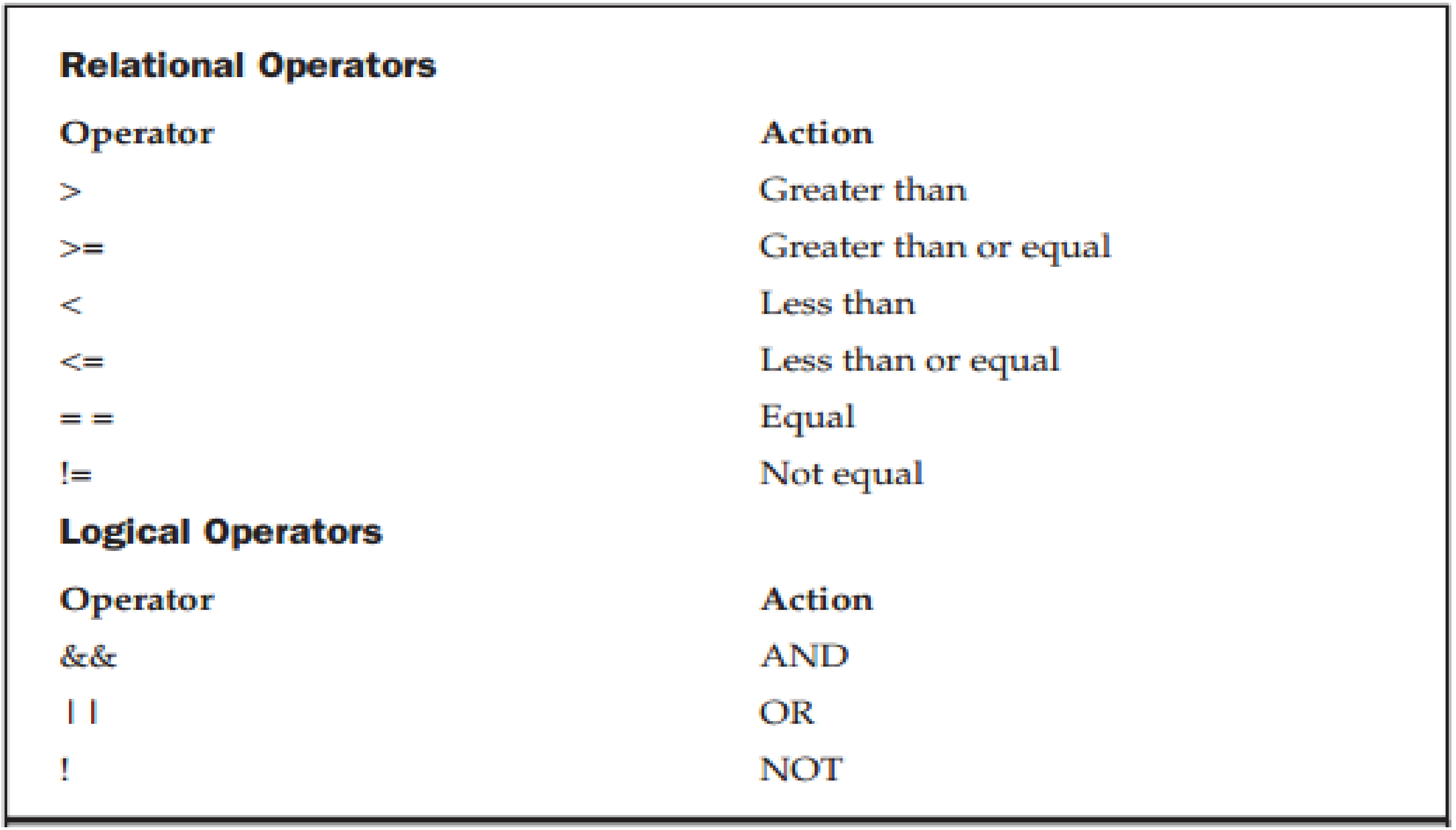
Write a program

- Convert a given number of seconds to hours, minutes and seconds then print them in the format: 5:17:9.

```
int seconds, minutes, hours;
cout << "Enter Seconds number ";
cin >> seconds;
hours = seconds / 3600;
minutes = seconds % 3600 / 60;
seconds = seconds % 3600 % 60;
cout << hours << ":" << minutes << ":" << seconds << endl;
```

Relational and Logical Operators

Expressions that use relational or logical operators return 0 for false and 1 for true.



Relational Operators

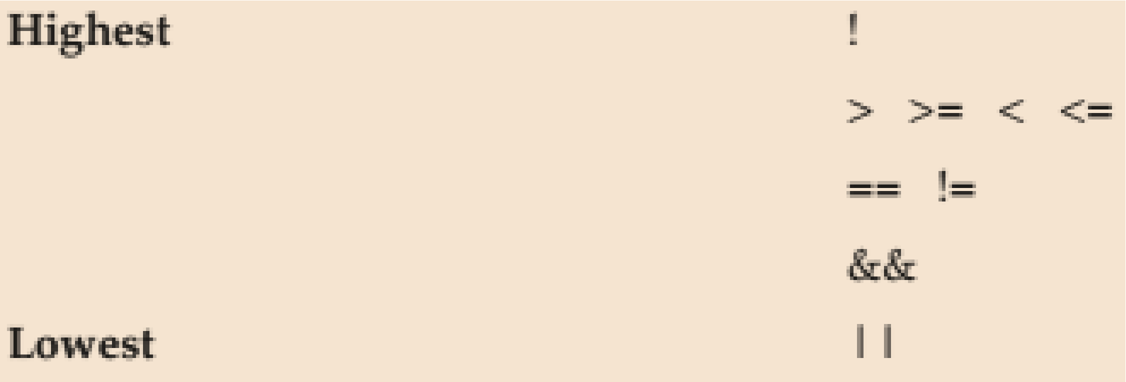
Operator	Action
>	Greater than
>=	Greater than or equal
<	Less than
<=	Less than or equal
= =	Equal
!=	Not equal

Logical Operators

Operator	Action
&&	AND
||	OR
!	NOT

Relational and Logical Operators

- Both the relational and logical operators are lower in precedence than the arithmetic operators.
- the relative precedence of the relational and logical operators:



Highest	!
	> >= < <=
	== !=
	&&
Lowest	||

Example

10>5 && !(10<9) || 3<=4 → True

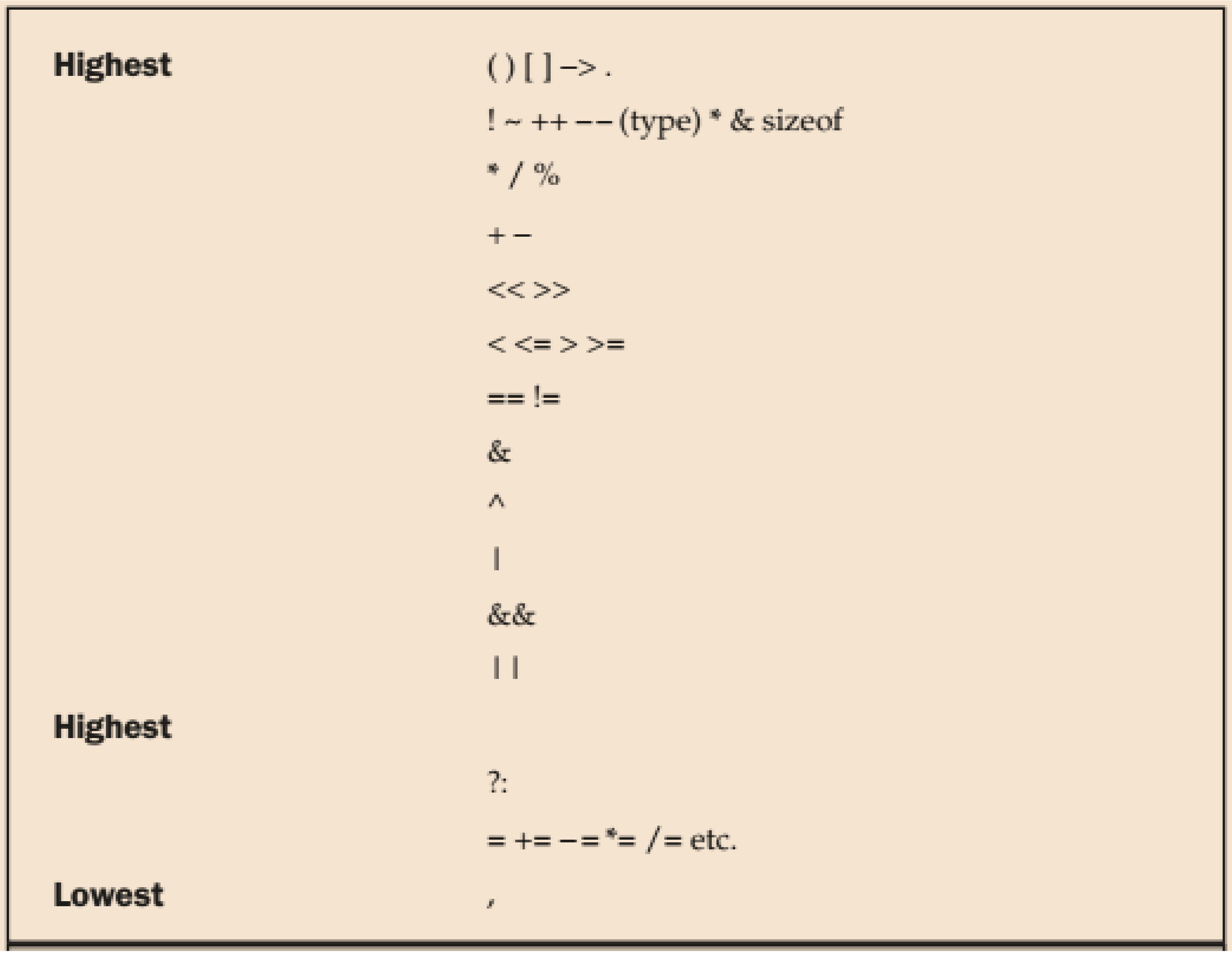
Show output

```cpp
#include<iostream>
using namespace std;
int main()
{
int y = 3;
int x = 10 < 5 && !(10 < 9) || ++y;
cout << "x=" << x << endl << "y=" << y;
system("Pause");
return 0;
}
```

One's complement (NOT)

Example

Original byte	→	0 0 1 0 1 1 0 0
After 1st complement	→	1 1 0 1 0 0 1 1
After 2nd complement	→	0 0 1 0 1 1 0 0
After Two’s Complement	→	1 1 0 1 0 1 0 0



Highest	( ) [ ] –> .
	! ~ ++ – – (type) * & sizeof
	* / %
	+ –
	<< >>
	< <= > >=
	== !=
	&
	^
	|
	&&
	| |
Highest	
	?:
	= += –= *= /= etc.
Lowest	,

Bool Data Type

- Two values: `true` and `false ( one or zero )`

```
#include <iostream>
#include<string>
using namespace std;
void main()
{
int x1 = 10, x2 = 20;
bool b1, b2;
b1 = x1 == x2;
b2 = x1 < x2;
cout << "b1 = " << b1 << "\n";
cout << "b2 = " << b2 << "\n";
system("Pause");
}
```

Go to

```
#include<iostream>
using namespace std;
void main()
{
int x, y;
a:
cout << "enter 2 numbers: ";
cin >> x >> y;
if (x > y)
cout << x << " is larger than " << y<<endl;
else
cout << y << " is larger than " << x<<endl;
goto a;
system("Pause");
}
```

Selection (Conditional) Statement

1) If statement

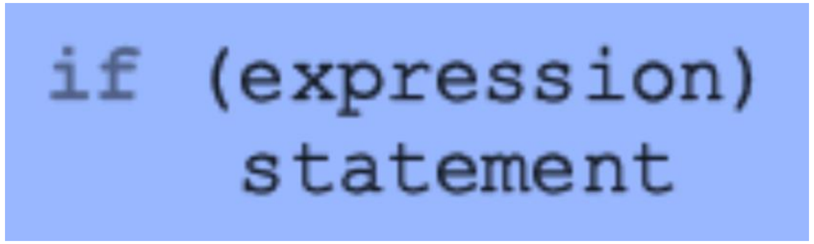
2) Switch cases

1) If Statement

- One way selection (If)
- Two-Way Selection (If …. else)
- Multiple Selections (Nested if)

One way selection

- General Form:



```
if (expression)
    statement
```

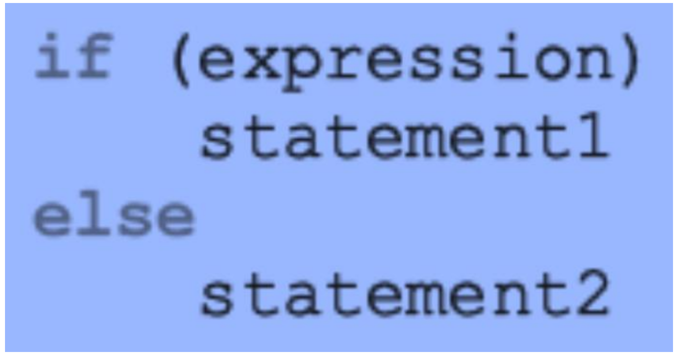
- The statement is executed if the value of the expression is true.

One way selection

```
#include<iostream>
using namespace std;
void main()
{
int number;
cout << "Enter number: ";
cin >> number;
if (number > 0)
cout << "number is Positive \n";
system("Pause");
}
```

Two-Way Selection

- General form:



```
if (expression)
     statement1
else
     statement2
```

- The statement1 is executed if the value of the expression is true. Otherwise, Statement2 is executed.

Two-Way Selection

```cpp
#include<iostream>
using namespace std;
void main()
{
int number;
cout << "Enter number: ";
cin >> number;
if (number >= 0)
cout << "number is Positive \n";
else
cout << "number is negative \n";
system("Pause");
}
```

Multiple Selections

- General form:

```
If (expression1)
     Statement1
else if (expression2)
            Statement2
else if (expression3)
            Statement3
.
.
else
       Statement
```

Multiple Selections

```
void main()
{
int number;
cout << "Enter number: ";
cin >> number;
if (number > 0)
cout << "number is Positive \n";
else if (number==0)
cout << "number is zero \n";
else
cout << "number is negative \n";
system("Pause");
}
```

Block of Statement

```
void main()
{
int number;
cout << "Enter number: ";
cin >> number;
if (number > 0)
{
cout << "number is Positive \n";
number += 5;
cout << "number become : " << number << endl;
}
system("Pause");
}
```

Examples

- Write a program that takes 3 integer from the user and prints the largest of these numbers.

```
int x, y, z;
cout << "Enter 3 numbers \n";
cin >> x >> y >> z;
if (x >= y&&x >= z)
cout <<"The largest number is:" <<x<<endl;
else if (y >= x&&y >= z)
cout << "The largest number is:" << y <<
endl;
else
cout << "The largest number is:" << z <<
endl;
```

Examples

- Write a program that reads a number and prints if it is odd or even.

```
int x;
cout << "Enter your number";
cin >> x;
if (x % 2 == 0)
cout << "number is even \n";
else
cout << "number is odd \n";
system("Pause");
```

Conditional Operator (?:)

General form:

expression1 ? expression2 : expression3

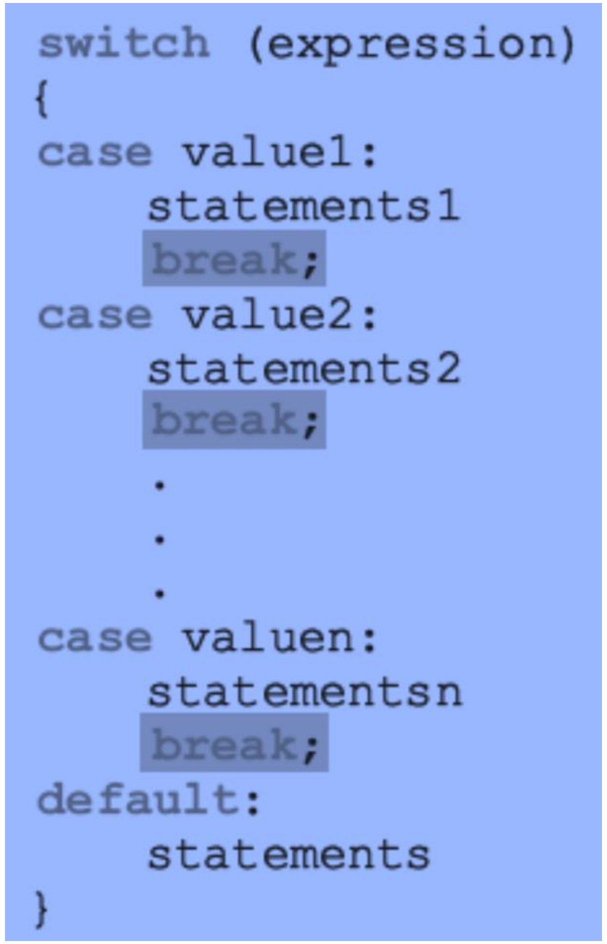
If expression1 is true apply expression2 else apply expression3

```
int x=10, y;
x>9 ? y=100 : y=200;
cout <<y<<endl;
```

```
int x=10, y;
if (x > 9)
y = 100;
else
y = 200;
cout << y<<endl;
```

2) Switch case

- General Form:



```
switch (expression)
{
case value1:
      statements1
      break;
case value2:
      statements2
      break;
      .
      .
      .
case valuen:
      statementsn
      break;
default:
      statements
}
```

- The expression must evaluate to a character or integer value.
- The value of expression is tested, in order, against the values of the constants in the case statements.
- When a match is found, the statement sequence associated with that case is executed until the break or the end of the switch statement is reached.
- The default statement is executed if no matches are found. The default is optional and, if it is not present, no action takes place if all matches fail.

Switch case Example1

```
char grade;
cout << "Enter Grade :";
cin >> grade;
switch (grade)
{
case 'A':
cout << "Excellent"<<endl;
break;
case 'B':
cout << "very Good" << endl;
break;
case 'C':
cout << "Good" << endl;
break;
case 'D':
cout << "Pass" << endl;
break;
case 'F':
cout << "Fail" << endl;
break;
default:
cout << "invalid grade!"
<< endl;
break;
}
```

Switch case Example2

```
char operation;
int num1, num2;
cout << "Enter 2 numbers :";
cin >> num1 >> num2;
cout << "choose one operation :
\n + add \n - subtract \n *
multiply \n / divide \n";
cin >> operation;
switch (operation)
{case '+':
cout << "sum = "<<num1+
num2<<endl;
break;
case '-':
cout << "difference = " << num1 -
num2 << endl;
break;
case '*':
cout << "multiply = " << num1 *
num2 << endl;
break;
case '/':
cout << "Division = " << num1 /
num2 << endl;
break;
default:
cout << "you entered invalid
operation !" << endl;
break;}
```

Print if number is odd or even using switch case

```
#include<iostream>
using namespace std;
void main()
{int num;
cout << "enter number";
cin >> num;
switch (num % 2)
{case 0:
cout << "number is even" << endl;
break;
case 1:
cout << "number is odd" << endl;
break;}
system("Pause");}
```

Example

Convert the following if statement to the Conditional Operator (?:)

```cpp
#include<iostream>
using namespace std;
void main()
{int x;
cout << "Enter x";
cin >> x;
if (x > 4)
x++;
else
x--;
cout << "x= " << x << endl;
system("Pause");}
```

Example

Convert the following if statement to the Conditional Operator (?:)

```
#include<iostream>
using namespace std;
int main()
{int x, y;
char op;
cout << "enter x, y and op:";
cin >> x >> y;
cin >> op;
if (op == '+')
cout << "sum = " << x + y << endl;
else if (op == '-')
cout << "sub = " << x - y << endl;
else if (op == '*')
cout << "Mul = " << x * y << endl;
else if (op == '/')
cout << "Div = " << x / y << endl;
else
cout << " invalide op\n";
system("Pause");}
```

```
#include<iostream>
using namespace std;
int main()
{
int x, y;
char op;
cout << "enter x, y and op:";
cin >> x >> y;
cin>> op;
op == '+' ? cout << "sum = " << x + y << endl
: op == '-' ? cout << "sub = " << x - y <<
endl : op == '*' ? cout << "Mul = " << x * y
<< endl : op == '/' ? cout << "Div = " << x /
y << endl : cout << " invalide op\n";
system("Pause");
return 0;}
```