Conditions in C++

اَللّٰهُمَّ اُرْزُقْنِى عِلْمًا نَافِعًا وَاسِعًا عَمِيْقًا

اَللّٰهُمَّ اُرْزُقْنِى رِزْقًا وَاسِعًا حَلَالًا طَيِّبًا
مُبَارَكًا مِنْ عِنْدِكَ

Why do we need Conditions?

When we add some kind of condition on some task, this is called conditional statement.

If there will be Good Weather then
we will go on Picnic.

Why do we need Conditions?

We speak many Conditional Statements daily in our life.

If you do my homework then i will offer you a Burger.

Why do we need Conditions?

As Programming solves Real World problems;
therefore, it also needs the Conditional Statements.

Review: Input/Output in C++

Let practice what we have learnt so far

Problem 01: Input/Output in C++

A Store has announced to give the 10% discount on the total purchase amount.
Write a Function that takes total purchase amount as input and returns the payable amount after discount.

Write Down Function ?

Steps: Towards Solution

- Think in terms of the function.
- Determine the function's input parameters, their data types and its return data type.
- While Deciding function header think function follows single responsibility principle

Solution

A Store has announced to give the 10% discount on the total purchase amount.
Write a Function that takes total purchase amount as input and returns the payable amount after discount.

```
float discount(float amount)
{
    float payable;
    payable = amount - (amount * 10)/100;
    return payable;
}
```

Input	Output
discount(1000);	900
discount(1545.5);	1390.95
discount(2000.14);	1800.13

Steps: Towards Solution

- Now, you have to write the main function.
- In the main you have to call your solution function. For passing it the parameters you first have to take input from the user and then call your function and then display the returned output.

Solution

```
int main()
{
    float amount, payable;
    cout << "Enter Purchase Amount: ";
    cin >> amount;
    payable = discount(amount);
    cout << "Payable Amount after discount: " << payable;
}
```

```
float discount(float amount)
{
    float payable;
    payable = amount - (amount * 10)/100;
    return payable;
}
```

Problem 02: Conditions in C++

A Store has announced to give the 10% discount on the total purchase amount only on Sunday.
Write a Function that takes Day and total purchase amount as input and returns the payable amount after discount.

Test Cases:

Input	Output
discount("Sunday", 1000);	900
discount("Monday", 1545.5);	1545.5
discount("Sunday", 2000.14);	1800.13

Steps: Towards Solution

- Now, we have to alter the flow of our function's execution and we want to give discount only on Sunday and on no other day.
- Whenever such condition arises we have to use IF statement.

```
float discount(float amount)
{
    float payable;
    payable = amount - (amount * 10)/100;
    return payable;
}
```

Solution: IF Statement

```
int main()
{
    float amount, payable;
    string day;
    cout << "Enter Shopping Day: ";
    cin >> day;
    cout << "Enter Purchase Amount: ";
    cin >> amount;
    payable = discount(day, amount);
    cout << "Payable Amount after discount: " << payable;
}
```

```
float discount(string day, float amount)
{
    float payable = amount;
    if (day == "Sunday")
    {
        payable = amount - (amount * 10) / 100;
    }
    return payable;
}
```

Problem 03: Conditions in C++

A Store has announced to give the 10% discount on the total purchase amount on Sunday and 5% on every other day.
Write a Function that takes Day and total purchase amount as input and returns the payable amount after discount.

Test Cases:

Input	Output
discount("Sunday", 1000);	900
discount("Monday", 1545.5);	1468.22
discount("Tuesday", 2000.14);	1900.13

Steps: Towards Solution

A Store has announced to give the 10% discount on the total purchase amount on Sunday and 5% on every other day.
Write a Function that takes Day and total purchase amount as input and returns the payable amount after discount.

Can we solve this problem with single IF statement?

Steps: Towards Solution

A Store has announced to give the 10% discount on the total purchase amount on Sunday and 5% on every other day.
Write a Function that takes Day and total purchase amount as input and returns the payable amount after discount.

```
if( condition1 ){
    do this
}
if( condition2 ){
    do this
}
```

We can solve this problem with multiple IF statements

Steps: Towards Solution

- Now, we have to add 2 conditions.
  1. If the day is Sunday
  2. If the day is not Sunday

Steps: Towards Solution

- Now, we have to add 2 conditions.
  1. If the day is Sunday
  2. If the day is not Sunday

Which comparison operator should we use for second condition?

Comparison Operators

Other than Equal (==), there are many comparison Operators.

Comparison Operators list

Comparison Operators	Description	Applicable on	Example
==	Equal to	Textual Data Numeric Data	if ("AB" == "AC") if (5 == 5)
!=	Not equal to	Textual Data Numeric Data	if ("AB" != "AC") if (5 != 3)
<	Less than	Numeric Data	if (2 < 4)
>	Greater Than	Numeric Data	if (4 > 4)
<=	Less than or equal to	Numeric Data	if (5 <= 90)
>=	Greater than or equal to	Numeric Data	if (66 >= 21)

Steps: Towards Solution

- Now, we have to add 2 conditions.
  1. If the day is Sunday
  2. If the day is not Sunday

Coming Back to the solution..

Solution: Multiple IF Statement

```
float discount(string day, float amount)
{
    float payable;
    if (day == "Sunday")
    {
        payable = amount - (amount * 10) / 100;
    }
    if (day != "Sunday")
    {
        payable = amount - (amount * 5) / 100;
    }
    return payable;
}
```

Solution: Multiple IF Statement

Conditions are Contradicting

```
float discount(string day, float amount)
{
    float payable;
    if (day == "Sunday")
    {
        payable = amount - (amount * 10) / 100;
    }
    if (day != "Sunday")
    {
        payable = amount - (amount * 5) / 100;
    }
    return payable;
}
```

Solution: IF-Else Statement

Conditions are Contradicting

```
float discount(string day, float amount)
{
    float payable;
    if (day == "Sunday")
    {
        payable = amount - (amount * 10) / 100;
    }
    else
    {
        payable = amount - (amount * 5) / 100;
    }
    return payable;
}
```

Which one is Better?

```
float discount(string day, float amount)
{
    float payable;
    if (day == "Sunday")
    {
        payable = amount - (amount * 10) / 100;
    }
    else
    {
        payable = amount - (amount * 5) / 100;
    }
    return payable;
}
```

If Else

```
float discount(string day, float amount)
{
    float payable;
    if (day == "Sunday")
    {
        payable = amount - (amount * 10) / 100;
    }
    if (day != "Sunday")
    {
        payable = amount - (amount * 5) / 100;
    }
    return payable;
}
```

Multiple If

Which one is Better?

```
float discount(string day, float amount)
{
    float payable;
    if (day == "Sunday")
    {
        payable = amount - (amount * 10) / 100;
    }
    else
    {
        payable = amount - (amount * 5) / 100;
    }
    return payable;
}
```

FALSE

If Else

```
float discount(string day, float amount)
{
    float payable;
    if (day == "Sunday")
    {
        payable = amount - (amount * 10) / 100;
    }
    if (day != "Sunday")
    {
        payable = amount - (amount * 5) / 100;
    }
    return payable;
}
```

TRUE

FALSE

Multiple If

Solution: IF-Else Statement

Main function will remain the same.

```
int main()
{
    float amount, payable;
    string day;
    cout << "Enter Shopping Day: ";
    cin >> day;
    cout << "Enter Purchase Amount: ";
    cin >> amount;
    payable = discount(day, amount);
    cout << "Payable Amount after discount: " << payable;
}
```

Multiple IF: Inevitable Cases

Sometimes, Conditions are not Contradicting but we have to make Different Decisions for Different Conditions.

Problem 04: Conditions in C++

A Store has announced to give the 10% discount on the total purchase amount on Sunday and 8% on Monday and 5% on Tuesday.
Write a Function that takes Day and total purchase amount as input and returns the payable amount after discount.

Test Cases:

Input	Output
discount("Sunday", 1000);	900
discount("Monday", 1545.5);	1421.86
discount("Tuesday", 2000.14);	1900.13
discount("Friday", 2000);	2000

Steps: Towards Solution

- Now, we have to add 3 conditions.
  1. If the day is Sunday
  2. If the day is Monday
  3. If the day is Tuesday

Solution: Multiple IF Statement

```
float discount(string day, float amount)
{
    float payable = amount;
    if (day == "Sunday")
    {
        payable = amount - (amount * 10) / 100;
    }
    if (day == "Monday")
    {
        payable = amount - (amount * 8) / 100;
    }
    if (day == "Tuesday")
    {
        payable = amount - (amount * 5) / 100;
    }
    return payable;
}
```

Review

Single IF Statement

```
if(condition){

}
```

Multiple IF Statement

```
if(condition1){

}
if (condition2){

}
```

IF-Else Statement

```
if(condition){

}
else{

}
```

Problem 05: Conditions in C++

A Store has announced to give the 10% discount on the total purchase amount on every Sunday of October only.
Write a Function that takes Day, Month and total amount as input and returns the payable amount after discount.

Test Cases:

Input	Output
discount("Sunday", "October", 4000);	3600
discount("Tuesday", "October", 4000);	4000
discount("Sunday", "March", 4000);	4000

Steps: Towards Solution

- Now, we have to give discount on 2 conditions.

  If the day is Sunday and the Month is October

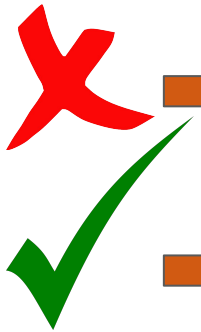
Solution: Will this Work?

```
float discount(string day, string month, float amount)
{
    float payable = amount;
    if (day == "Sunday")
    {
        payable = amount - (amount * 10) / 100;
    }
    if (month == "October")
    {
        payable = amount - (amount * 10) / 100;
    }
    return payable;
}
```

Solution: Will this Work?

Lets Dry Run the first Test Case.

discount("Tuesday", "October", 4000) → 4000



```
float discount(string day, string month, float amount)
{
    float payable = amount;
    if (day == "Sunday")
    {
        payable = amount - (amount * 10) / 100;
    }
    if (month == "October")
    {
        payable = amount - (amount * 10) / 100;
    }
    return payable;
}
```

3600

Failed

Solution: Will this Work?

This type of conditions are called simultaneous conditions that need to be checked in parallel

```
float discount(string day, string month, float amount)
{
    float payable = amount;
    if (day == "Sunday")
    {
        payable = amount - (amount * 10) / 100;
    }
    if (month == "October")
    {
        payable = amount - (amount * 10) / 100;
    }
    return payable;
}
```

Solution: Will this Work?

discount("Tuesday", "October", 4000) → 4000

```
float discount(string day, string month, float amount)
{
    float payable = amount;
    if (day == "Sunday")
    {
        if (month == "October")
        {
            payable = amount - (amount * 10) / 100;
        }
    }
    return payable;
}
```

4000

Passed

Solution: Nested IF Statements

When we have another IF block inside another IF block, we called it Nested IF Block.

```
float discount(string day, string month, float amount)
{
    float payable = amount;
    if (day == "Sunday")
    {
        if (month == "October")
        {
            payable = amount - (amount * 10) / 100;
        }
    }
    return payable;
}
```

Solution: Nested IF Statements

Can we give the solution with single IF statement?

```
float discount(string day, string month, float amount)
{
    float payable = amount;
    if (day == "Sunday")
    {
        if (month == "October")
        {
            payable = amount - (amount * 10) / 100;
        }
    }
    return payable;
}
```

Solution: IF Statement & logical Operators

The solution with single IF statement. Here && is used to check two statements simultaneously.

```
float discount(string day, string month, float amount)
{
    float payable = amount;
    if (day == "Sunday" && month == "October")
    {
        payable = amount - (amount * 10) / 100;
    }
    return payable;
}
```

Logical Gates

Before moving to the solution, lets see the Truth tables of some Logical Gates you have already studied.

Logical Gates: AND Gate

Let X and Y are two separate conditions and if we combine these conditions with AND operator then the result will be true only when both conditions are true.

Logical Gates: AND Gate

Let X and Y are two separate conditions and if we combine these conditions with AND operator then the result will be true only when both conditions are true

For example:

X = if today is raining

Y = if today is sunday

X and Y will return true if and only if there is raining and day is sunday

Logical Gates: AND Gate

lets see the Truth tables of AND Logical Gate is

X	Y	X AND Y (X ^ Y)
False	False	
False	True	
True	False	
True	True	

Logical Gates: AND Gate

lets see the Truth tables of AND Logical Gate is

X	Y	X AND Y (X ^ Y)
False	False	False
False	True	False
True	False	False
True	True	True

Logical Gates: OR Gate

Let X and Y are two separate conditions and if we combine these conditions with OR operator then the result will be true when any of these conditions are true

Logical Gates: OR Gate

Let X and Y are two separate conditions and if we combine these conditions with OR operator then the result will be true when any of these conditions are true

For example:

X = if today is raining

Y = if today is sunday

***X OR Y* will return true if and only if there is raining or day is sunday**

Logical Gates: OR Gate

lets see the Truth tables of AND Logical Gate is

X	Y	X AND Y (X ^ Y)	X OR Y (X V Y)
False	False	False	
False	True	False	
True	False	False	
True	True	True	

Logical Gates: OR Gate

lets see the Truth tables of AND Logical Gate is

X	Y	X AND Y (X ^ Y)	X OR Y (X V Y)
False	False	False	False
False	True	False	True
True	False	False	True
True	True	True	True

Logical Gates: Not Gate

lets see the Truth tables of AND Logical Gate is

X	Y	X AND Y (X ^ Y)	X OR Y (X V Y)	NOT X (~ X)
False	False	False	False	
False	True	False	True	
True	False	False	True	
True	True	True	True	

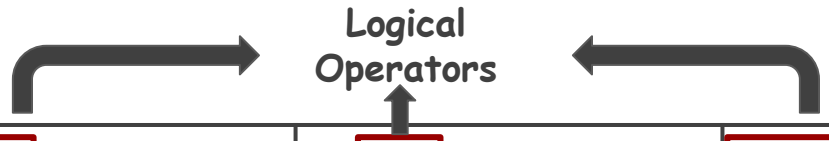
Logical Gates: Not Gate

lets see the Truth tables of AND Logical Gate is

X	Y	X AND Y (X ^ Y)	X OR Y (X V Y)	NOT X (~ X)
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

Logical Gates

lets see the Truth tables of AND Logical Gate is



X	Y	X AND Y (X ^ Y)	X OR Y (X V Y)	NOT X (~ X)
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

Logical Operators:

In C++, we can also use these logical operators.

Operator
AND
OR
Not

Logical Operators:

In C++, we can also use these logical operators.

Operator	In C++
AND	&&
OR	||
Not	!

Logical Operators:

In C++, we can also use these logical operators.

Operator	In C++	Example Suppose x = 3
AND	&&	x < 1 && x < 5
OR	||	x < 1 || x < 5
Not	!	! (x < 1 && x < 5)

Logical Operators:

In C++, we can also use these logical operators.

Operator	In C++	Example Suppose x = 3	Intermediate Result
AND	&&	x < 1 && x < 5	False && True
OR	||	x < 1 || x < 5	False || True
Not	!	! (x < 1 && x < 5)	! (False && True)

Logical Operators:

In C++, we can also use these logical operators.

Operator	In C++	Example Suppose x = 3	Intermediate Result	Final Result
AND	&&	x < 1 && x < 5	False && True	False
OR	||	x < 1 || x < 5	False || True	True
Not	!	! (x < 1 && x < 5)	! (False && True)	True

Problem 06: Conditions in C++

A Store has announced to give the 10% discount on the total purchase amount on every sunday or Month is October.
Write a Function that takes Day, Month and total amount as input and returns the payable amount after discount.

Test Cases:

Input	Output
discount("Sunday", "October", 4000);	3600
discount("Tuesday", "October", 4000);	3600
discount("Sunday", "November", 4000);	3600

Solution: IF Statement & logical Operators

```
float discount(string day, string month, float amount)
{
    float payable = amount;
    if (day == "Sunday" || month == "October")
    {
        payable = amount - (amount * 10) / 100;
    }
    return payable;
}
```

Problem 07: Conditions in C++

A Store has announced to give the 10% discount on the total purchase amount on every sunday of Month October or March or August.
Write a Function that takes Day, Month and total amount as input and returns the payable amount after discount.

Test Cases:

Input	Output
discount("Sunday", "August", 4000);	3600
discount("Tuesday", "October", 4000);	4000

Solution: IF Statement & logical Operators

```
float discount(string day, string month, float amount)
{
    float payable = amount;
    if (day == "Sunday" && (month == "October" || month == "March" || month == "August"))
    {
        payable = amount - (amount * 10) / 100;
    }

    return payable;
}
```

Problem 08: Conditions in C++

A Store has announced to give the 10% discount on the total purchase amount on every sunday and Month is October, March and August and 5% discount on the total purchase amount of every monday of November and December.
Write a Function that takes Day, Month and total amount as input and returns the payable amount after discount.
Test Cases:

Input	Output
discount("Sunday", "August", 4000);	3600
discount("Tuesday", "October", 4000);	4000
discount("Monday", "November", 4000);	3800

Solution: IF Statement & logical Operators

```
float discount(string day, string month, float amount)
{
    float payable = amount;
    if (day == "Sunday" && (month == "October" || month == "March" || month == "August"))
    {
        payable = amount - (amount * 10) / 100;
    }
    if (day == "Monday" && (month == "November" || month == "December"))
    {
        payable = amount - (amount * 5) / 100;
    }
    return payable;
}
```

What will be the Output?

```
1   #include <iostream>
2   using namespace std;
3
4   main(){
5       int money = 0;
6       string meal = "fruit";
7                       1       1       0       0       0
8       if(meal == "fruit" || meal == "sandwich" && money >= 2){
9           cout << "Lunch being delivered" << endl;
10      }
11      else{
12          cout << "Cannot deliver Lunch" << endl;
13      }
    }
```

Logical Operators

What will be the Output?

```
1   #include <iostream>
2   using namespace std;
3
4   main(){
5       int money = 0;
6       string meal = "fruit";
7                       1       1       0       0       0
8       if(meal == "fruit" || meal == "sandwich" && money >= 2)
9           cout << "Lunch being delivered" << endl;
10      }
11      else{
12          cout << "Cannot deliver Lunch" << endl;
13      }
    }
```

Logical Operators

Which one is correct?

```
1   #include <iostream>
2   using namespace std;
3
4   main(){
5       int money = 0;
6       string meal = "fruit";
7
8       if(meal == "fruit" || meal == "sandwich" && money >= 2){
9           cout << "Lunch being delivered" << endl;
10      }
11      else{
12          cout << "Cannot deliver Lunch" << endl;
13      }
    }
```

Precedence Order

Before answering Which one is correct, we must know about the precedence of logical operators.

Precedence Order	Operator	In C++
1	Not	!
2	AND	&&
3	OR	||

What will be the Output?

```
1   #include <iostream>
2   using namespace std;
3
4   main(){
5       int money = 0;
6       string meal = "fruit";
7
8       if(meal == "fruit" || meal == "sandwich" && money >= 2){
9           cout << "Lunch being delivered" << endl;
10      }
11      else{
12          cout << "Cannot deliver Lunch" << endl;
13      }
    }
```

Conclusion

- C++ supports three types of logical operators (AND, OR and NOT).
- Logical operators are used to combine multiple conditions so that these conditions can be applied in a single if statement. The result of the operation of a logical operator is a boolean value either true or false.
- The AND operator is used to combine multiple conditional statements and it returns true only when the conditions around it are true.
- OR operator returns true when any one or both of the conditions are true.
- NOT operator reverses the result.
- The order of precedence between logical operators is
  NOT
  AND
  OR

Learning Outcome

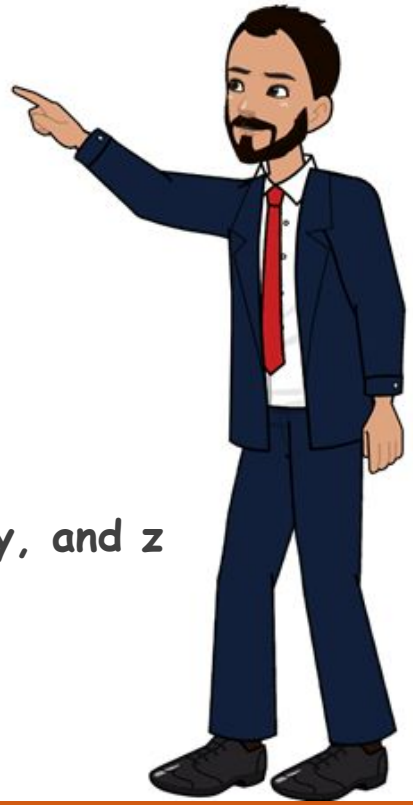
In this lecture, we learnt how to write a C++ program for complex conditional statements with multiple Boolean expressions using AND, OR and NOT logical operators while considering the Precedence Rules.

Self Assessment

After execution of the following code, what is stored in the number? (All variables are of type int.)
Hint: use x=3, y=5, and z=7 as sample values.

```
if (y > z) {
if (x > y)
     number = x;
else
     number = y;
}
else {
     if (x > z)
          number = x;
     else
          number = z;
}
```

a. the smallest value of x, y, and z
b. the largest value of x, y, and z
c. smaller of x and y
d. larger of x and z
e. randomly selected value from x, y, and z



Self Assessment

What will be the output of the following:

```
int m = 8, n = 12;
if (!(m > 5) && (n < 10 || m > 10))
    cout << "A";
else
    cout << "B";
```

```
int a = 5, b = 7, c = 9;
if (a > b && b < c || a + b == c)
    cout << "P";
else
    cout << "Q";
```

Self Assessment

What will be the output of the following:

```
bool x = true, y = false, z = true;
if ((x && y) || (!z))
    cout << "X";
else
    cout << "Y";
```

```
bool p = true, q = false, r = true;
if (!(!p || q && r))
    cout << "Hello";
else
    cout << "World";
```

Self Assessment

Solve Following Programs

1. Take the age and name of three brothers as input and display the younger brother's name.

2. Write a program that asks the user to input three different values and then find out the largest using nested if.

Self Assessment

Solve Following Programs

Ali is a teacher, he needs a program which helps him to compile his class results. He has 5 subjects (English, Math, Chemistry, Social Science and Biology) marked in detail. Program asks the user to enter 5 subjects' marks including student name and displays the total marks, percentage, grade (by percentage) and student name. Every subject has a total 100 marks. Grading policy details are mentioned below in table

90-100 percentage	A+
80-90 percentage	A
70-80 percentage	B+
60-70 percentage	B
50-60 percentage	C
40-50 percentage	D
Below 40 percentage	F

Self Assessment

Solve Following Programs

Write a program that asks the user for 3 different integers. If one of those integers is equal to or greater than 50, print out "One of Value is too large."

Write a program that asks the user which province they live in. If the province isn't "Sindh", print out "You should come visit Sindh sometime!".

Self Assessment

Solve Following Question

Insert parentheses into the following expression to show how operator precedence groups operands :

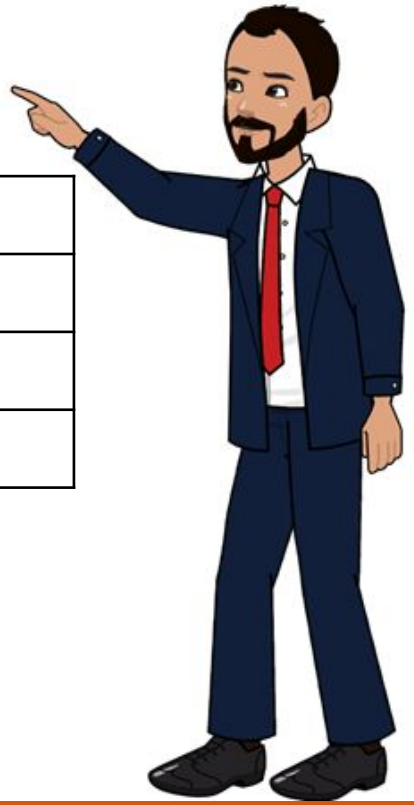
```
a > b && 45 <= sum || sum < a + b && d > 90
```

Don't change the meaning of the expression; use parentheses to make the order of evaluation clear.

Self Assessment

Fill Following Table

A || B && C	means	A || (B && C)
A && B || C && D	means	
A && B && C || D	means	
!A && B || C	means	



Self Assessment

Solve Following Programs

Write the code which asks for a login.
If the visitor enters "Admin", then prompt for a password. If the input is an empty line – show “Canceled”. If it is another string, then show “I don’t know you”.
The password is checked as follows:
If it equals “TheMaster”, then show “Welcome!”,
Another string – show “Wrong password”
For an empty string or cancelled input, show “Canceled”
as shown in the diagram in the next slide

Self Assessment

Solve Following Programs