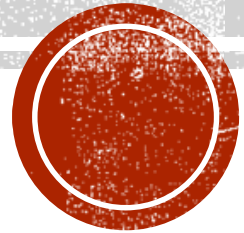
INSTRUCTIONS OF SELECTIONS
IF, IF-ELSE



OBJECTIVES

1. Introduction
2. Blocks of instructions
3. If
4. If…else
5. Nested if, Nested if…else

1. INTRODUCTION

- Definition of Instructions of selection
- Why they are needed?

2. BLOCK OF INSTRUCTIONS

- If zero or more instructions are enclosed between two braces they form a block of instructions (each instruction is terminated by ;)

```
{
x=2;
}
```

```
{ x=2;}
```

```
{x=2;
Y=5;}
```

```
{x=2; Y=5;}
```

```
{cin>>x;
 Y=5;
Z=y+x;}
```

```
{cin>>x; Y=5; Z=y+x;}
```

2. BLOCK OF INSTRUCTIONS

- If zero or more instructions are enclosed between two braces they form a block of instructions (each instruction is terminated by ;)

```
{
x=2;
}
```

```
{x=2;
Y=5;}
```

```
{cin>>x;
 Y=5;
Z=y+x;}
```

{ x=2;}

{x=2; Y=5;}

{cin>>x; Y=5; Z=y+x;}

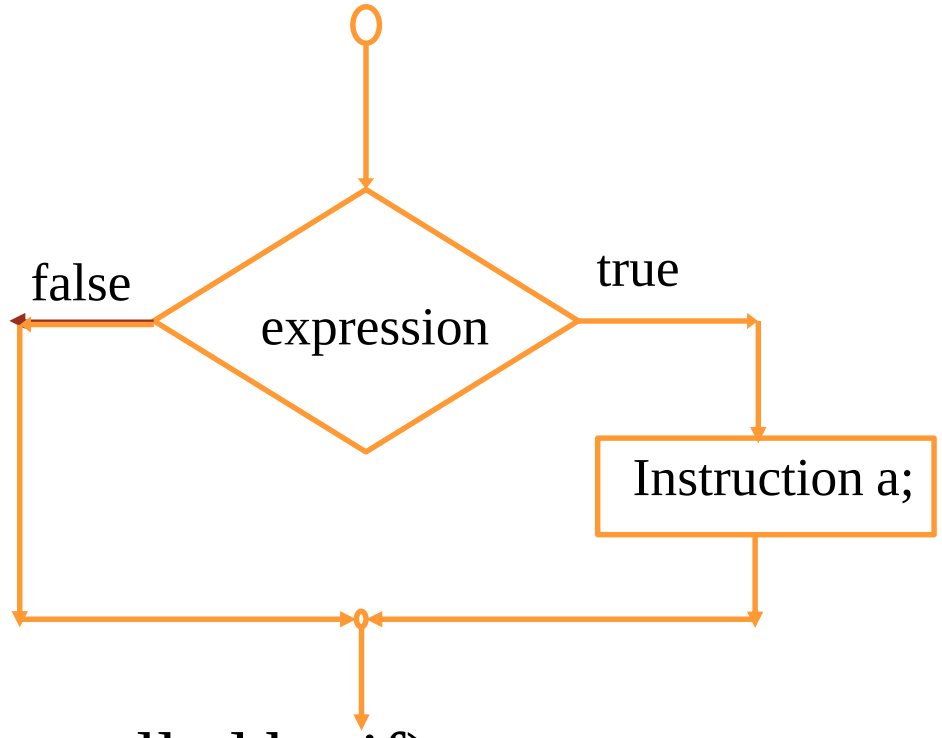
1. { } ✓
2. { i=1; } ✓
3. { ; } ✓
4. { i=5; k=3 } ✗

3. INSTRUCTION OF SELECTION IF

- Syntax

if *(expression)*
instruction a;

Instruction controlled by if



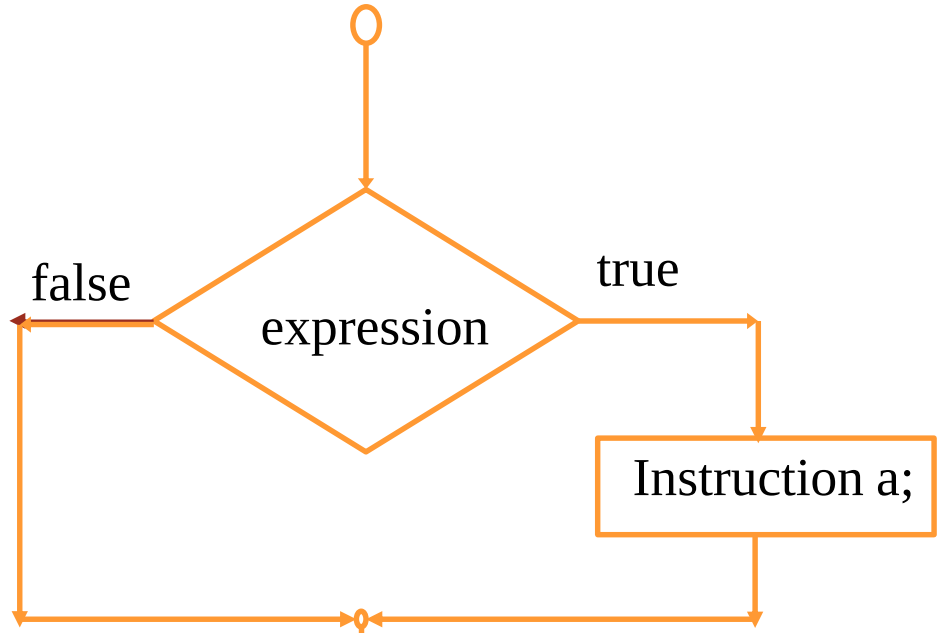
- *Instruction a* is the instruction belong to if (controlled by if)
- *Instruction a* is executed only in the case of the value of *expression* is equal to true (condition of if).

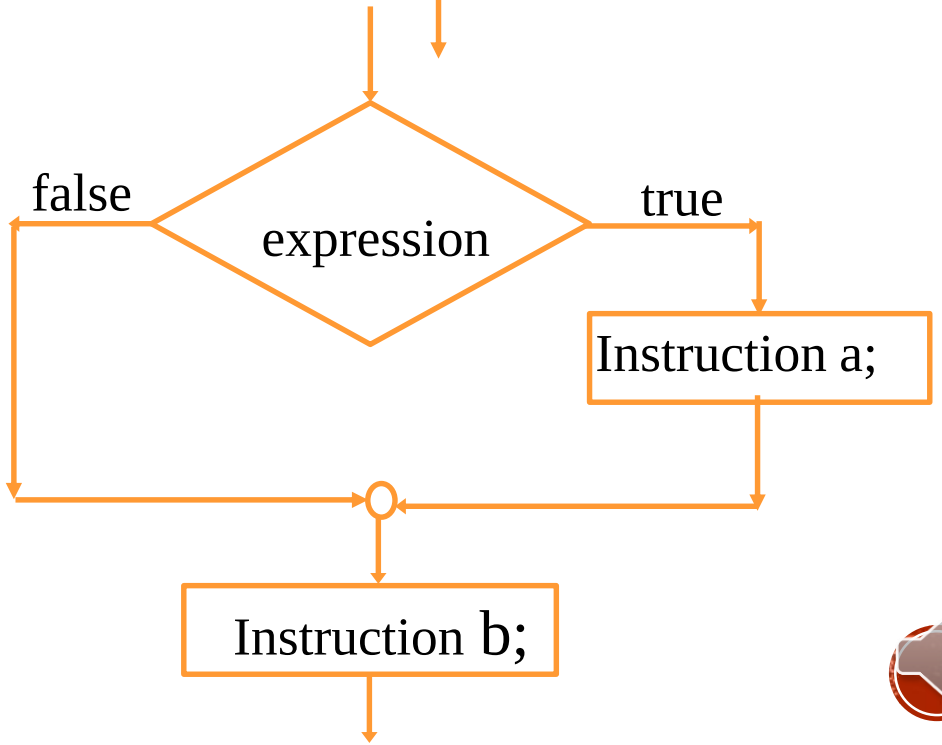
3. INSTRUCTION OF SELECTION IF

- Syntax

if *(expression)*
instruction a; ← Instruction controlled by if

if *(expression)*
instruction a; ← One Instruction controlled by if
instruction b; ← Next instruction



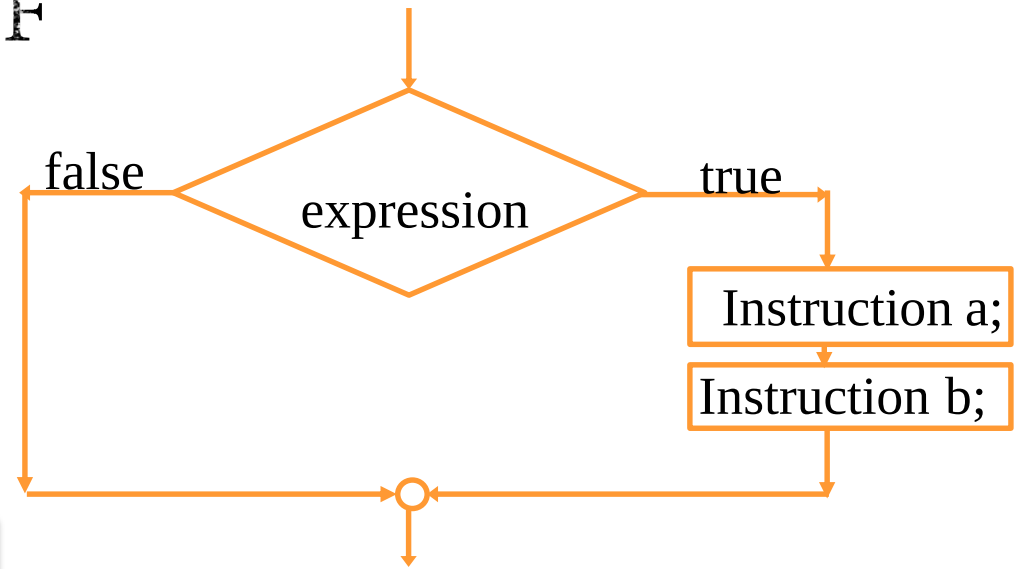


3. INSTRUCTION OF SELECTION IF

- Syntax

```
if (expression)
{
instruction a;
instruction b;
}
```

Block of Instructions controlled by if



- If more than one instruction are needed to be controlled by the instruction ***if***, the *instruction a* is replaced by the block of instructions that encloses the controlled instructions.
- In our shown example the instruction ***if*** controls a block of instructions that contains the *instruction a* and *the instruction b.*

3. INSTRUCTION OF SELECTION IF

if *(expression)*
  instruction a;

- ***expression*** can be:

3. INSTRUCTION OF SELECTION IF

if *(expression)*
instruction a;

- ***expression*** can be :
  1. Boolean value: result of comparison operators: >, <, >=, <=, ==, !=

3. INSTRUCTION OF SELECTION IF

if *(expression)*
instruction a;

- ***expression*** can be :
  1. Boolean value: result of comparison operators: >, <, >=, <=, ==, !=

Example :

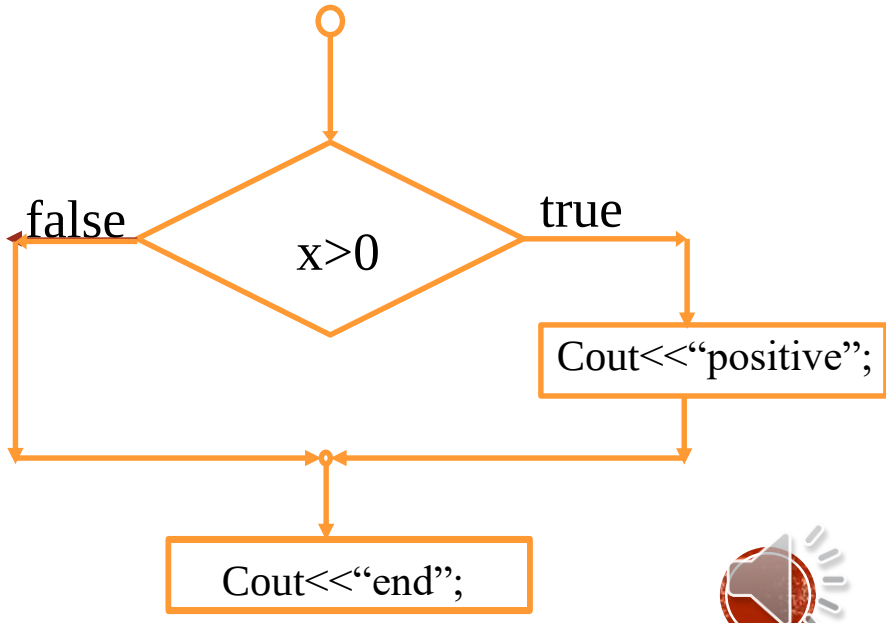
```
if (x>0)
cout << "positive\n";
cout<<"end";
```

Screen x=5;

```
positive
end
```

Screen x=-3

```
end
```



3. INSTRUCTION OF SELECTION IF

if *(expression)*
  instruction a;

- ***expression*** can be :
  1. Boolean value
  2. Boolean expression: result of operators &&, ||, and !

3. INSTRUCTION OF SELECTION IF

if *(expression)*
  instruction a;

- ***expression*** can be :
  1. Boolean value
  2. <u>Boolean expression:</u> result of operators &&, ||, and !

<u>Example :</u>

```
if (x>10 && x<12)
cout << " grade A\n";
cout<<"end";
```

Screen x=11;

```
grade A
end
```

Screen x=20;

```
end
```

3. INSTRUCTION OF SELECTION IF

if *(expression)*

instruction a;

- ***expression*** can be :
  1. Boolean value
  2. Boolean expression: result of operators &&, ||, and !

Example :

```
if (x>y || x>z)
     cout<<“ x\n”;
if ((x>y && y>z ) || (x==2))
     cout <<“ x is max or x = 2\n”;
cout<<“end”;
```

x=2; y=5; z=10;

Screen

```
x is max or x=2
end
```

x=9; y=5; z=10;

Screen

```
x
end
```

x=15; y=5; z=10;

Screen

```
x
x  is max or x=2
end
```

3. INSTRUCTION OF SELECTION IF

if *(expression)*

instruction a;

- ***expression*** can be :
  1. Boolean value
  2. Boolean expression
  3. Numerical value (int, float, …):
     - Different than zero: considered as true
     - Zero: considered as false

3. INSTRUCTION OF SELECTION IF

if *(expression)*
  instruction a;

- ***expression*** can be :
  1. Boolean value
  2. Boolean expression
  3. Numerical value (int, float, …):
     - Different than zero: considered as true
     - Zero: considered as false

Example :

```
if (x)
cout << "Hello\n";
cout<<"End";
```

Screen x=5

```
Hello
End
```

Screen x=-2

```
Hello
End
```

Screen x=0

```
End
```

3. INSTRUCTION OF SELECTION IF

if *(expression)*

instruction a;

- ***expression*** can be :
  1. Boolean value
  2. Boolean expression
  3. Numerical value
  4. Numerical expression
     - If the result different than zero: considered as true
     - If the result is equal to Zero: considered as false

3. INSTRUCTION OF SELECTION IF

if *(expression)*
  instruction a;

- ***expression*** can be :
  1. Boolean value
  2. Boolean expression
  3. Numerical value
  4. <u>Numerical expression</u>
     - If the result different than zero: considered as true
     - If the result is equal to Zero: considered as false

<u>Example :</u>

```
if (x-y)
cout << "Hello\n";
cout<<"End";
```

Screen x=5, y=2

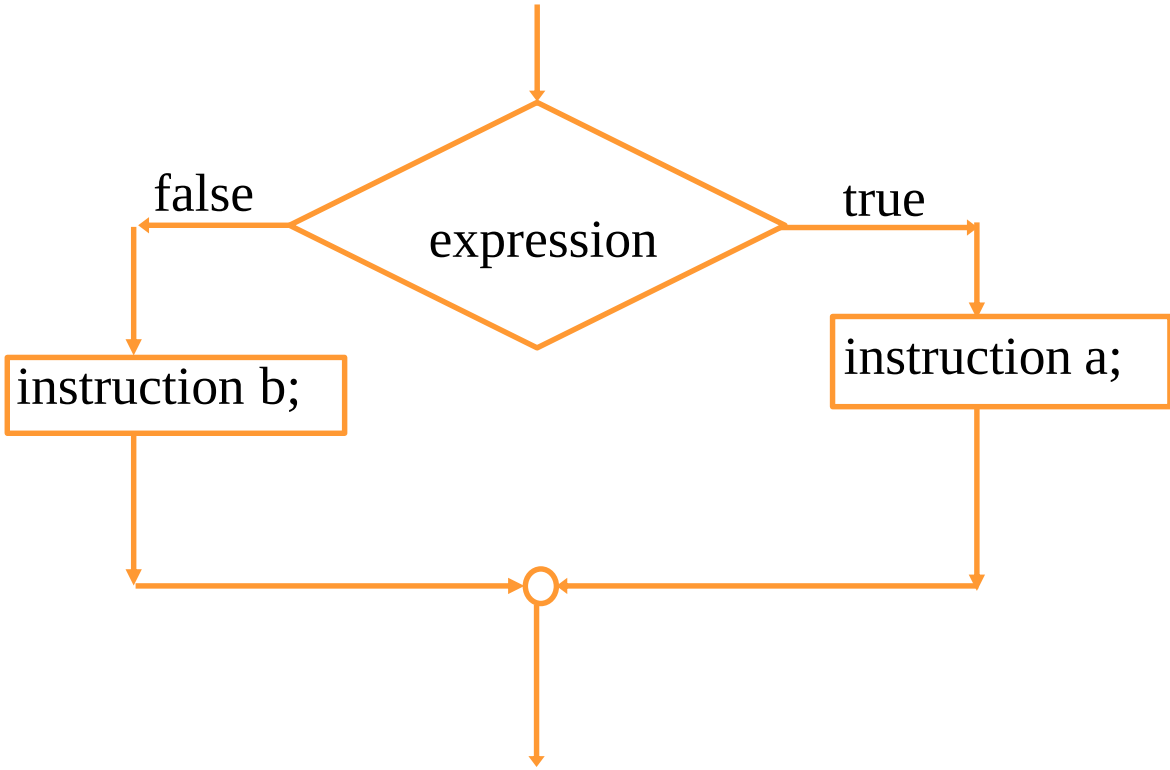
```
Hello
End
```

Screen x=2, y=2

```
End
```

4- IF….ELSE

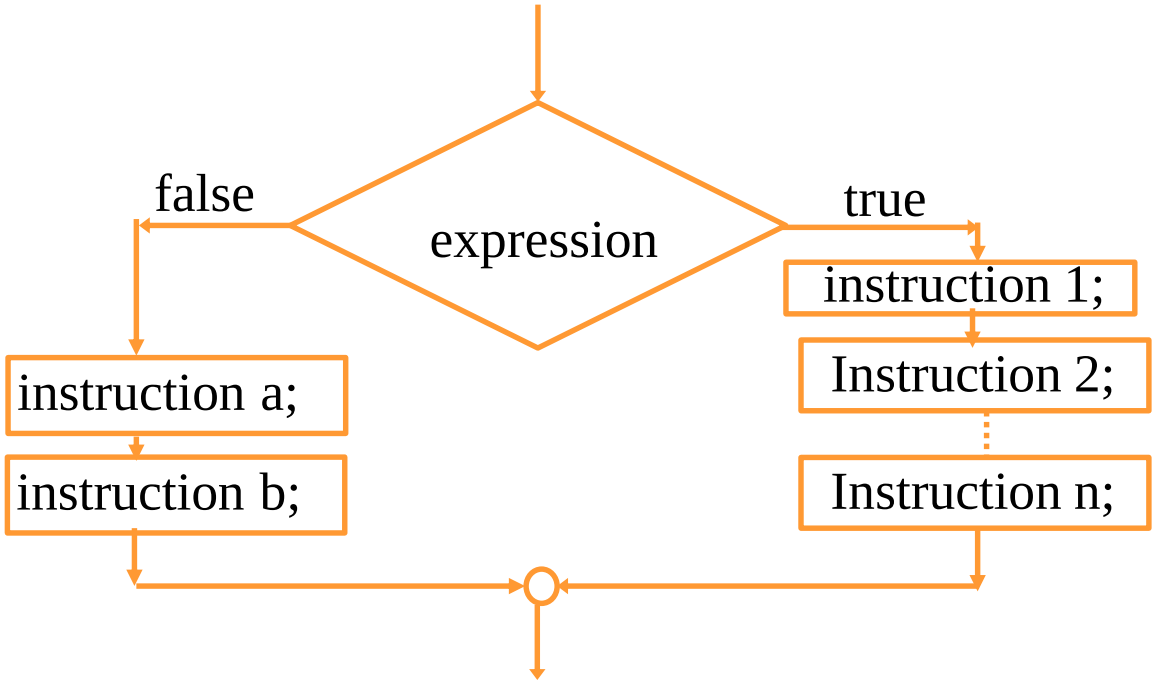
if *(expression)*
instruction a;
else
instruction b;



4- IF….ELSE

```
if (expression)
instruction a;
else
instruction b;
```

```
if (expression)
 {
    instruction 1;
    instruction 2;
   ...
    instruction n;
 }
else
{
    instruction a ;
    instruction b ;
}
```



5- NESTED IF, NESTED IF…ELSE

```
if ( <expr1> )
    <bloc1>
else
    if (<expr2 >)
        <bloc2>
    else
        if (<expr3 >)
            <bloc3>
        else
                <bloc4>
```

*In C/C++ an **else** is always linked to the last **if** which does not have an else.*

5- NESTED IF, NESTED IF…ELSE

<bloci> is executed in case of :

```
if ( <expr1> )
    <bloc1> ←
else
    if (<expr2 >)
        <bloc2> ←
    else
        if (<expr3 >)
            <bloc3> ←
        else
                <bloc4> ←
```

5- NESTED IF, NESTED IF…ELSE

<bloci> is executed in case of :

```
if ( <expr1> )
    <bloc1>          ← <expr1> is true
else
    if (<expr2 >)
        <bloc2>      ← <expr1> is false and <expr2> is true
    else
        if (<expr3 >)
            <bloc3>  ← <expr1> is false &&<expr2> is false && <expr3> is true
        else
            <bloc4>  ← <expr1> is false &&<expr2> is false && <expr3> is false
```

5- NESTED IF, NESTED IF…ELSE

```
if ( <expr1> )
    <bloc1>

if (<expr2 > && !<expr1>)
        <bloc2>

if (<expr3 >&& !<expr2 > && !<expr1>)
        <bloc3>
if(!< expr3 >&& !<expr2 > && !<expr1>)
                <bloc4>
```

<bloci> is executed in case of :

```
if ( <expr1> )
    <bloc1>                 <expr1> is true
else
    if (<expr2 >)
        <bloc2>             <expr1> is false and <expr2> is true
    else
        if (<expr3 >)
            <bloc3>         <expr1>  is false &&<expr2> is
                                false && <expr3> is true
        else
            <bloc4>         <expr1> is false &&<expr2>
                                is false && <expr3> is false
```

5- NESTED IF, NESTED IF…ELSE (EXAMPLE)

```
if (N>0)
     if (A>B)
          MAX = A;
     else
          MAX = B;
```

```
if (N>0)
{
     if (A>B)
          MAX = A;
}
else
     MAX = B;
```

5- NESTED IF, NESTED IF…ELSE (EXAMPLE)

```
if (N>0)
    if (A>B)
        MAX = A;
    else
        MAX = B;
```

2 different codes!!

```
if (N>0)
{
    if (A>B)
        MAX = A;
}
else
    MAX = B;
```

To avoid confusion and to force a certain interpretation of an expression, it is recommended to use braces {}

5- NESTED IF, NESTED IF…ELSE (EXAMPLE)

```
if(A>B) cout<< " first choice";
else
      if (A>10) cout<< "second choice";
      else
            if(B<10) cout<<"third choice";
            else cout<<"fourth choice";
```

5- NESTED IF, NESTED IF…ELSE (EXAMPLE)

```
if(A>B) cout<< " first choice";
else
      if (A>10) cout<< "second choice";
      else
            if(B<10) cout<<"third choice";
            else cout<<"fourth choice";
```

```
if(A>B)
      cout<< " first choice";
else
      {
         if (A>10)
                  cout<< "second choice";
          else
                  if(B<10)
                        cout<<"third choice";
                  else
                        cout<<"fourth choice";
      }
```

5- NESTED IF, NESTED IF…ELSE (EXAMPLE)

```
if(A>B)
      cout<< " first choice";
else
      {
         if (A>10)
                  cout<< "second choice";
          else
                  if(B<10)
                        cout<<"third choice";
                  else
                        cout<<"fourth choice";
      }
```

- For which values of A and B do we get on the screen the results:
  - first choice,
  - second choice,
  - third choice
  - fourth choice
- For which values of A and B do not get an answer on the screen?

5- NESTED IF, NESTED IF…ELSE (EXAMPLE)

```
if(A>B)
      cout<< " first choice";
else
      {
         if (A>10)
                cout<< "second choice";
          else
                if(B<10)
                      cout<<"third choice";
                else
                      cout<<"fourth choice";
      }
```

- For which values of A and B do we get on the screen the results:
  - first choice: A>B
  - second choice, A<=B && A>10
  - third choice A<=B && A<=10 && B<10
  - fourth choice A<=B && A<=10 && B>=10
- For which values of A and B do not get an answer on the screen?

Always we get an answer on the screen.