

## Branching Statement I:

Apart from the reserved word **exit**, there are a few reserved words that allow you jumping from one line to another line or a scope to another scope in the same block.

### Label...goto:

The **label...goto** statement allows you to marked line in the source code with specific labels/identifiers and to use the reserved word **goto** to jump to that specific line and run code from that line, thus the execution sequence will change.

*Avoid the use of this statement, as it only increases complexity in reading the program.*

### Syntax:

**Label** *Label identifier1, Label identifierN;*

**Label** *0.1.2....N: { N ≤ 9999 } {0..N won't work}*

### Usage: (In part of code)

```
Label_identifier1: statement;
...
goto Label_identifier1;
```

### Example: Change execution order.

```
Label 1;
Begin
Goto 1;
Write(2);
1:Write(3);
End.
```

Output 3

```
Label 1,2,3;
Begin
goto 2;
1:Write(2); goto 3;
2:Write(3); goto 1;
3:End.
```

Output 32

### Example:

Perform like a looping

```
var i:integer;
Label 1,2,3;
Begin
i:=1;
1:write(i);
if i=10 then goto 3;
2: inc(i); goto 1;
3: writeln;
End.
```

For I:=1 to 10 do write(i);writeln;

```
var i:integer;
Label 1;
Begin
i:=1;
1: i:=i*2;
if i<1000 then goto 1;
writeln(i);
End.
```

Repeat I:=I\*2 until I>=1000;

```
var i:integer;
label 1,2,3;
begin
i:=1;
1: if i>=1000 then goto 3;
2: i:=i*2; goto 1;
3: writeln(i);
end.
```

While I<1000 do I:=I\*2;

### Example: perform recursion

```
const lev=2;
ball=4;
var stk:array[1..lev] of integer;
top,now,i:integer;
label recur,recur2;
begin
for i:=1 to lev do stk[i]:=0;
top:=0;
now:=1;
recur:
if (lev+1=now) then
begin
for i:=1 to lev do write(stk[i],' ');
writeln; {Continue on R.H.S}
```

```
dec(now);
dec(top);goto recur2; end
else
begin
for i:=1 to ball do
begin
inc(top);
stk[top]:=i;
inc(now);
goto recur;
recur2:
i:=stk[now];
end;
dec(now);
```

```
dec(top);
if now<>0 then goto recur2;
end;
end.
```

Output all permutations of drawing 2 balls from 4 balls with repetition.

```
(1,1) (1,2) (1,3) (1,4)
(2,1) (2,2) (2,3) (2,4)
(3,1) (3,2) (3,3) (3,4)
(4,1) (4,2) (4,3) (4,4)
```

*Pascal allows you only jump within the same block. This is seldom tested in HKOI.*

## Branching Statement II:

You may skip the **label...goto** statement if it's too difficult.  
 However, the rests are very important.  
 You must make sure yourself understands them.

### Break:

It is a reserved word that allows you to end the nearest and innermost loop immediately.

### Example:

<pre>var k,j:integer; begin for k:=1 to 2 do for j:=1 to 2 do begin writeln(k,j); break; end; end.</pre>	<pre>var k,j:integer; begin for k:=1 to 2 do while true do begin writeln(k); break; end; end.</pre>	<pre>Var k,j:integer; Begin For k:=1 to 2 do For j:=1 to 2 do begin if k&gt;j then break; writeln(k,j); end; End.</pre>	<pre>Var k,j:integer; Begin For k:=1 to 2 do For j:=1 to 2 do begin if k=j then break; writeln(k,j); end; End.</pre>
<pre>11 21</pre>	<pre>1 2</pre>	<pre>11 12</pre>	<pre>21</pre>

### Continue:

It is a reserved word that allows you to skip the current iteration of the nearest loop.

### Example:

<pre>var k:integer; begin for k:=1 to 5 do begin if k=3 then continue; write(k); end; end.</pre>	<pre>var k:integer; begin for k:=1 to 5 do begin if k&lt;3 then continue; write(k); end; end.</pre>	<pre>var j,k:integer; begin for j:=1 to 5 do for k:=1 to 5 do if k&gt;j then continue else write(k); end.</pre>
<pre>1245</pre>	<pre>345</pre>	<pre>112123123412345</pre>

*These two reserved words can only exist in a looping.  
 Otherwise, a syntax error occurs.*