PASCAL - SUBPROGRAM CALL BY REFERENCE

http://www.tutorialspoint.com/pascal/pascal_call_by_reference.htm

The **call by reference** method of passing arguments to a subprogram copies the address of an argument into the formal parameter. Inside the subprogram, the address is used to access the actual argument used in the call. This means that changes made to the parameter affect the passed argument.

In order to pass the arguments by reference, Pascal allows to define **variable parameters**. This is done by preceding the formal parameters by the keyword **var**. Let us take the example of the procedure *swap* that swaps the values in two variables and reflect the change in the calling subprogram.

```
procedure swap(var x, y: integer);
var
   temp: integer;

begin
   temp := x;
   x:= y;
   y := temp;
end;
```

Next, let us call the procedure *swap* by passing **values by reference** as in the following example −

```
program exCallbyRef;
var
   a, b : integer;
(*procedure definition *)
procedure swap(var x, y: integer);

var
   temp: integer;

begin
   temp := x;
   x:= y;
   y := temp;
end;

begin
   a := 100;
   b := 200;
   writeln('Before swap, value of a : ', a );
   writeln('Before swap, value of b : ', b );

   (* calling the procedure swap  by value   *)
   swap(a, b);
   writeln('After swap, value of a : ', a );
   writeln('After swap, value of b : ', b );
end.
```

When the above code is compiled and executed, it produces the following result −

```
Before swap, value of a : 100
Before swap, value of b : 200
After swap, value of a : 200
After swap, value of b : 100
```

Which shows that now the procedure swap has **changed the values in the calling program**.