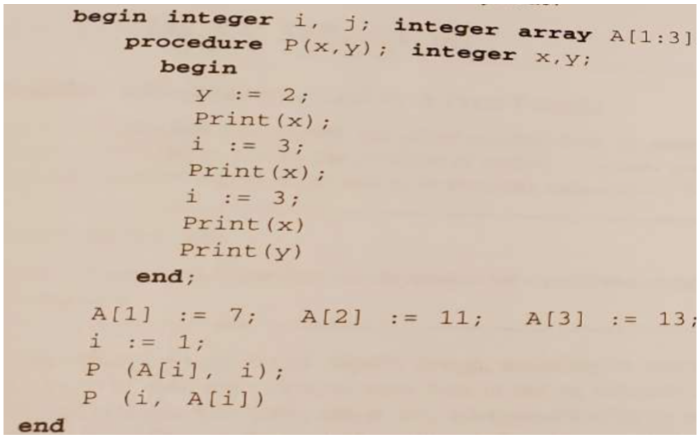
**1: (45 points)Problem 10 in page 141, but after adding another case "e", where both parameters x and y are passed by value-result. For each case, show the changes to the values of all involved variables (i.e., array A and i, in the caller block and x and y in the callee P) as you step executing in the code.**
**Change the array values to be:**
**A[1] = 10**
**A[2] = 5**
**A[3] = 15**



```
begin integer i, j; integer array A[1:3]
   procedure P(x,y); integer x,y;
     begin
       y := 2;
       Print(x);
       i := 3;
       Print(x);
       i := 3;
       Print(x)
       Print(y)
     end;
   A[1] := 7;   A[2] := 11;   A[3] := 13;
   i := 1;
   P (A[i], i);
   P (i, A[i])
end
```

### 1a: x is passed by value and y is passed by value

P(A[i], i):
A[] = [10, 5, 15], i = 1, x = 10, y = 1
A[] = [10, 5, 15], i = 1, x = 10, y = 2
A[] = [10, 5, 15], i = 3, x = 10, y = 2
Prints:
10
10
10
2
P(i, A[i]):
A[] = [10, 5, 15], i = 3, x = 3, y = 15
A[] = [10, 5, 15], i = 3, x = 3, y = 2
Prints:
3
3
3
2

### 1b: x is passed by value and y is passed by name

P(A[i], i):
A[] = [10, 5, 15], i = 1, x = 10, y = 1

A[] = [10, 5, 15], i = 2, x = 10, y = 2
A[] = [10, 5, 15], i = 3, x = 10, y = 3
Prints:
10
10
10
3
P(i, A[i]):
A[] = [10, 5, 15], i = 3, x = 3, y = 15
A[] = [10, 5, 15], i = 3, x = 3, y = 2
Prints:
3
3
3
2

## 1c: x is passed by name and y is passed by value

P(A[i], i):
A[] = [10, 5, 15], i = 1, x = 10, y = 1
A[] = [10, 5, 15], i = 1, x = 10, y = 2
A[] = [10, 5, 15], i = 3, x = 15, y = 1
Prints:
10
10
15
1
P(i, A[i]):
A[] = [10, 5, 15], i = 3, x = 3, y = 15
A[] = [10, 5, 15], i = 3, x = 3, y = 2
Prints:
3
3
3
2

## 1d: x is passed by name and y is passed by name

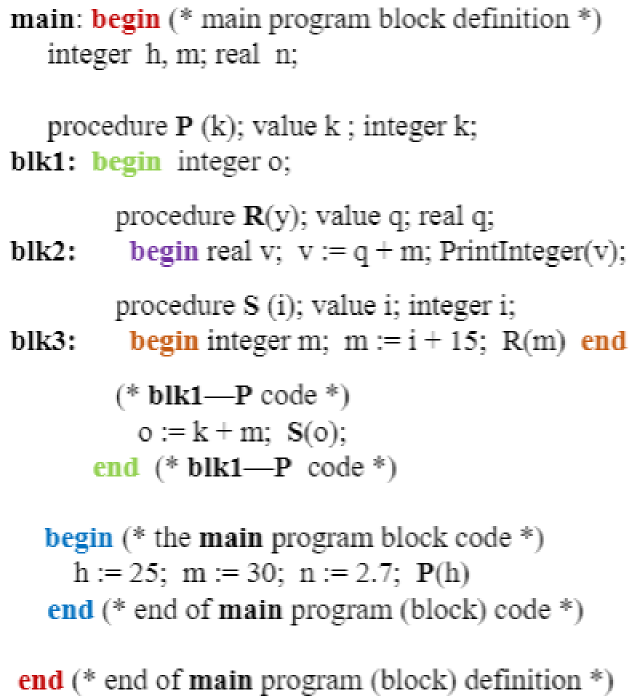
P(A[i], i):
A[] = [10, 5, 15], i = 1, x = 10, y = 1
A[] = [10, 5, 15], i = 2, x = 5, y = 2
A[] = [10, 5, 15], i = 3, x = 15, y = 3
Prints:
5
15
15
3

P(i, A[i]):
A[] = [10, 5, 15], i = 3, x = 3, y = 15
A[] = [10, 5, 15], i = 3, x = 3, y = 2
3
3
3
2

### 1e: x is passed by value-result and y is passed by value result

P(A[i], i):
A[] = [10, 5, 15], i = 1, x = 10, y = 1
A[] = [10, 5, 15], i = 1, x = 10, y = 2
A[] = [10, 5, 15], i = 3, x = 10, y = 2
Prints:
10
10
10
2
P(i, A[i]):
A[] = [10, 5, 15], i = 2, x = 2, y = 5
A[] = [10, 5, 15], i = 2, x = 2, y = 2
A[] = [10, 5, 15], i = 3, x = 2, y = 2
Prints:
2
2
2
2

### 2a: (20 points) Given the following ALGOL code:

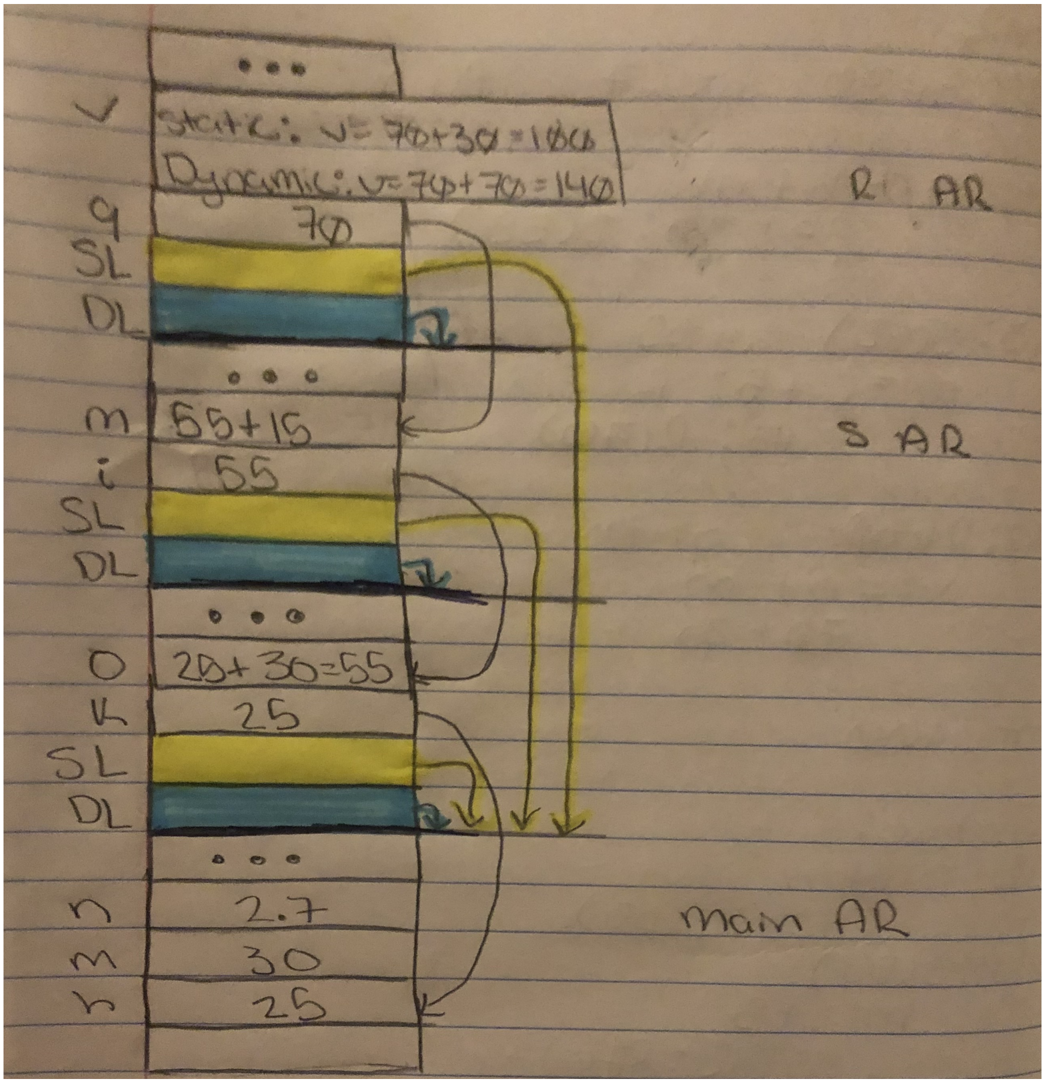


```
main: begin (* main program block definition *)
    integer  h, m; real  n;

    procedure P (k); value k ; integer k;
blk1:  begin  integer o;

           procedure R(y); value q; real q;
blk2:        begin real v;  v := q + m; PrintInteger(v); end (* blk2*);

           procedure S (i); value i; integer i;
blk3:        begin integer m;  m := i + 15;  R(m)  end (*end blk3*);

           (* blk1—P code *)
             o := k + m;  S(o);
         end  (* blk1—P  code *)

    begin (* the main program block code *)
       h := 25;  m := 30;  n := 2.7;  P(h)
    end (* end of main program (block) code *)

 end (* end of main program (block) definition *)
```

**What would be the printed value of "v", in blk2 above, in each of the following scoping mechanisms: i) static? ii) dynamic?**
**You must show the picture of the system stack upon the execution of the "PrintInteger(v)"**

---

i) Static: v = 100
ii) Dynamic v = 140



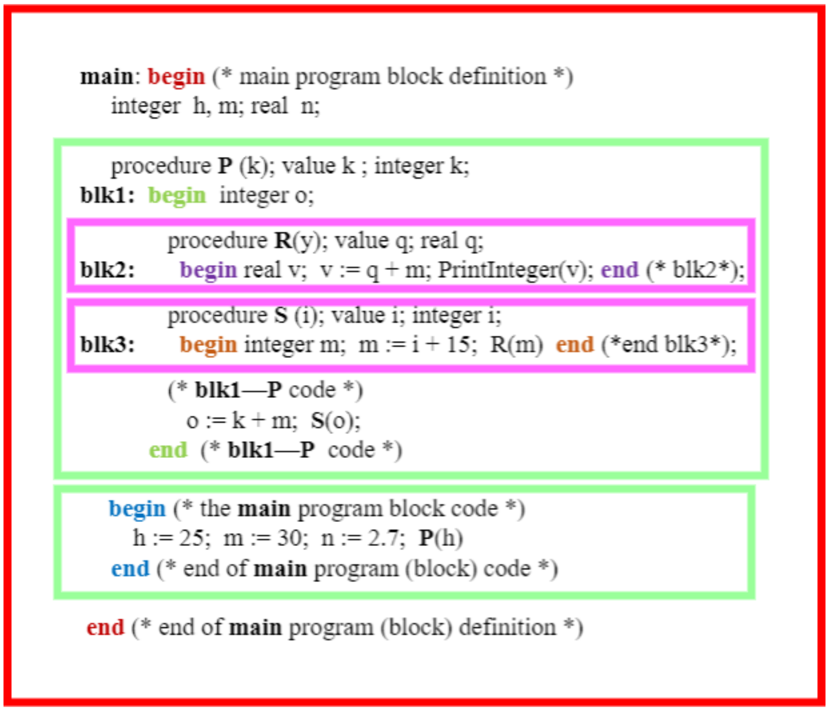
NOTE: In the above diagram the missing label For the AR should be labeled P. i.e. "P AR"

---

**2b: (6 points)Draw the contour diagram of the above ALGOL program and answer the following questions, based on that diagram:**

- **Can we call P() inblk2?**
- **Can we call S() in the main blockmain?**
- **Can we call "blk3" inR()?**

---

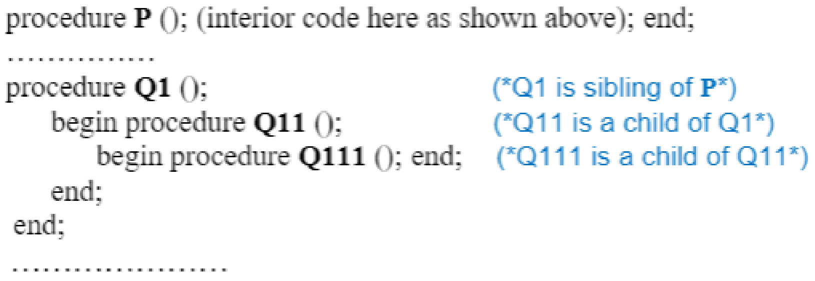


```
  main: begin (* main program block definition *)
     integer  h, m; real  n;

     procedure P (k); value k ; integer k;
  blk1:  begin  integer o;

         procedure R(y); value q; real q;
  blk2:     begin real v;  v := q + m; PrintInteger(v); end (* blk2*);

         procedure S (i); value i; integer i;
  blk3:     begin integer m;  m := i + 15;  R(m)  end (*end blk3*);

         (* blk1—P code *)
           o := k + m;  S(o);
       end  (* blk1—P  code *)

     begin (* the main program block code *)
       h := 25;  m := 30;  n := 2.7;  P(h)
     end (* end of main program (block) code *)

   end (* end of main program (block) definition *)
```

i) Yes, we can call P() in blk2 in both static and dynamic scoping
ii) No, we cannot call S() in the main block for both static and dynamic scoping
iii) Yes, we can call blk3 in R() in for both static and dynamic scoping

---

**2c: (12 + 7 + 10 points)To improve our code, we have created a sibling procedure to the P(), called Q1(), which is declared immediately after P(). Q1() will have an internal procedure Q11() that declares an internal procedure Q111(). Remember, both P() and Q1() are siblings in the "main" block (program).Here the layout of the code explained in part "c", in the above paragraph:**



```
procedure P (); (interior code here as shown above); end;
...............
procedure Q1 ();                              (*Q1 is sibling of P*)
    begin procedure Q11 ();                   (*Q11 is a child of Q1*)
        begin procedure Q111 (); end;         (*Q111 is a child of Q11*)
    end;
 end;
......................
```

**Now answer the following questions:**

- **List the names of all procedures that are visible to be called from within the code of: Q1(), Q111(), R(), and P() assuming static scoping.**
- **If we assume dynamic scoping, can we still use the contour diagram to answer part "i" above? Justify your yes/no answer.**
- **In case of dynamic scoping, is there a scenario that allows the calling of procedure Q111() from within S()? ONLY when you answer YES, show such scenario.**

i) Q1() → main(), Q11(), P()
Q111() → main(), Q11(), Q1(), P()
R() → main(), P(), S(), Q1()
P() → main(), Q1(), R(), S()

ii) No because the static structure of the previous contour diagram cannot describe the run time behavior of the of the new program.
iii) No