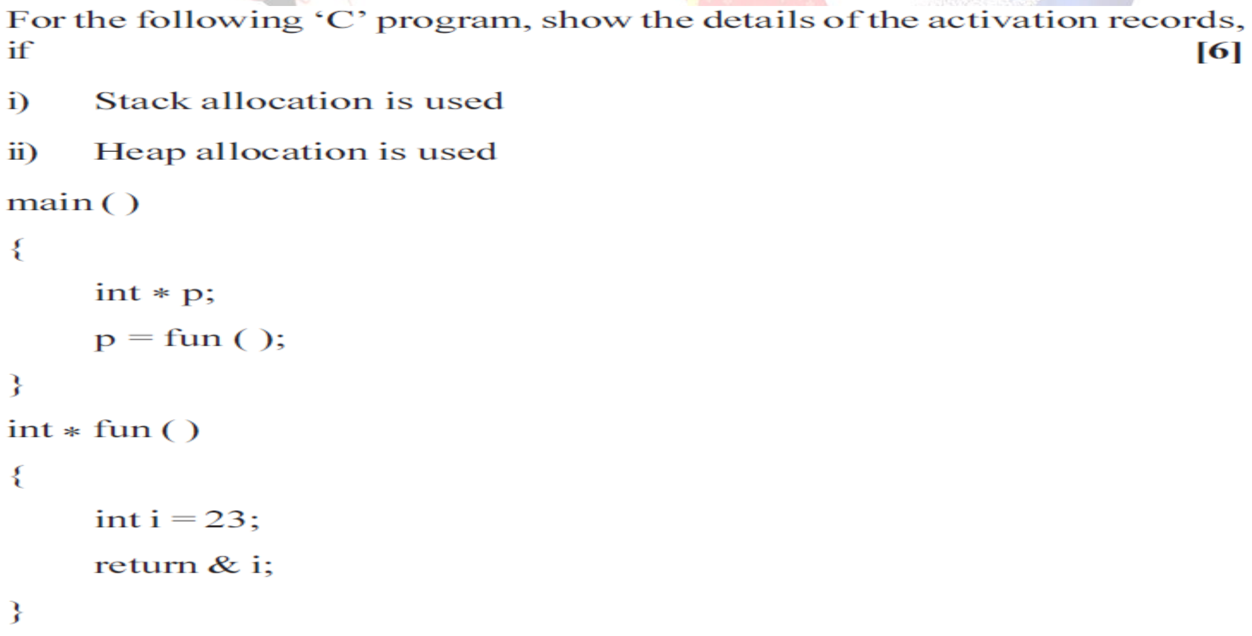
Pune Vidyarthi Griha’s

COLLEGE OF ENGINEERING, NASHIK – 4

COMPUTER ENGINEERING DEPARTMENT

Subject : COMPILER **ASSIGNMENT NO – 04** **Unit : IV**

1. Explain in detail about Run Time Storage Allocation.
   OR Discuss storage organization and allocation strategies.
2. Describe the Static and Dynamic Scope with example.
   OR Compare Static scope & Dynamic scope with example.
3. What is an Activation Record. Explain each of field in details?
4. What are two approaches of implementing Dynamic scope? Give the different between the two.
5. Discuss in brief various parameter passing techniques?
6. Explain issues related to nested procedures.
7. Explain Run Time Management variable length data.
8. Discuss various data structure for symbol table.
9. Explain Display and how display is used to access Non-local data?
10. Compare and contrast static storage management and dynamic storage management.
11.



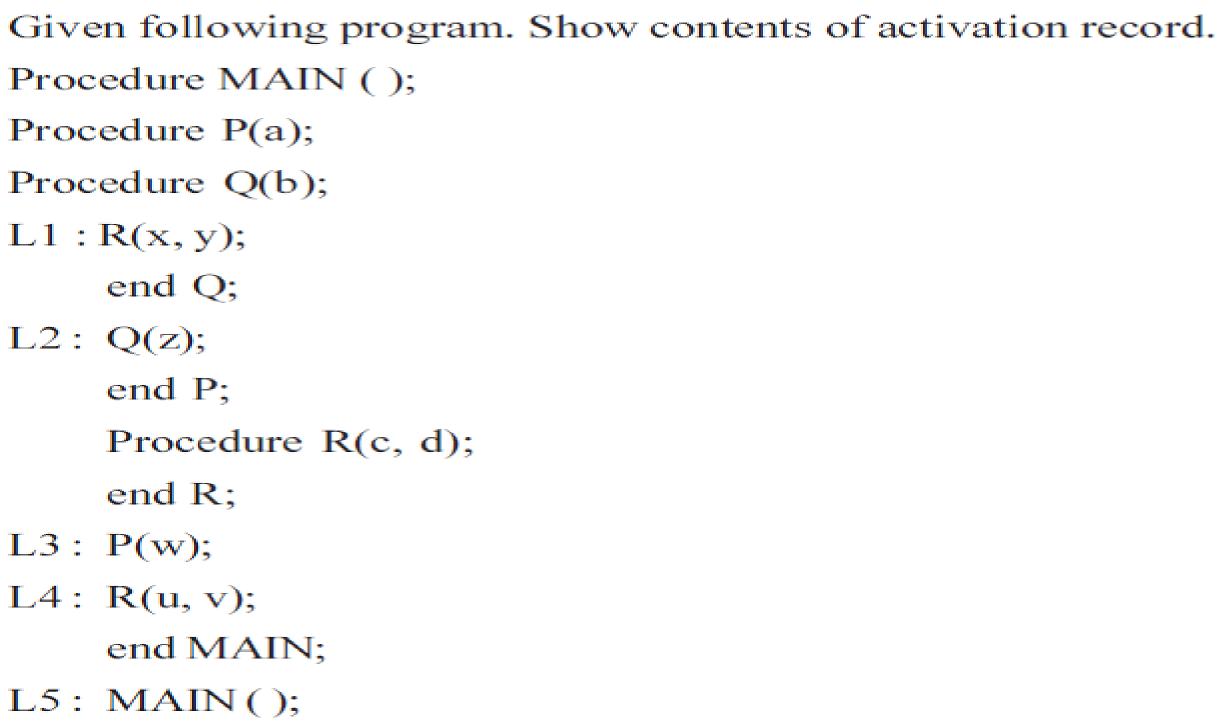
For the following ‘C’ program, show the details of the activation records, if **[6]**

i) Stack allocation is used

ii) Heap allocation is used

```
main ( )
{
        int * p;
        p = fun ( );
}
int * fun ( )
{
        int i = 23;
        return & i;
}
```

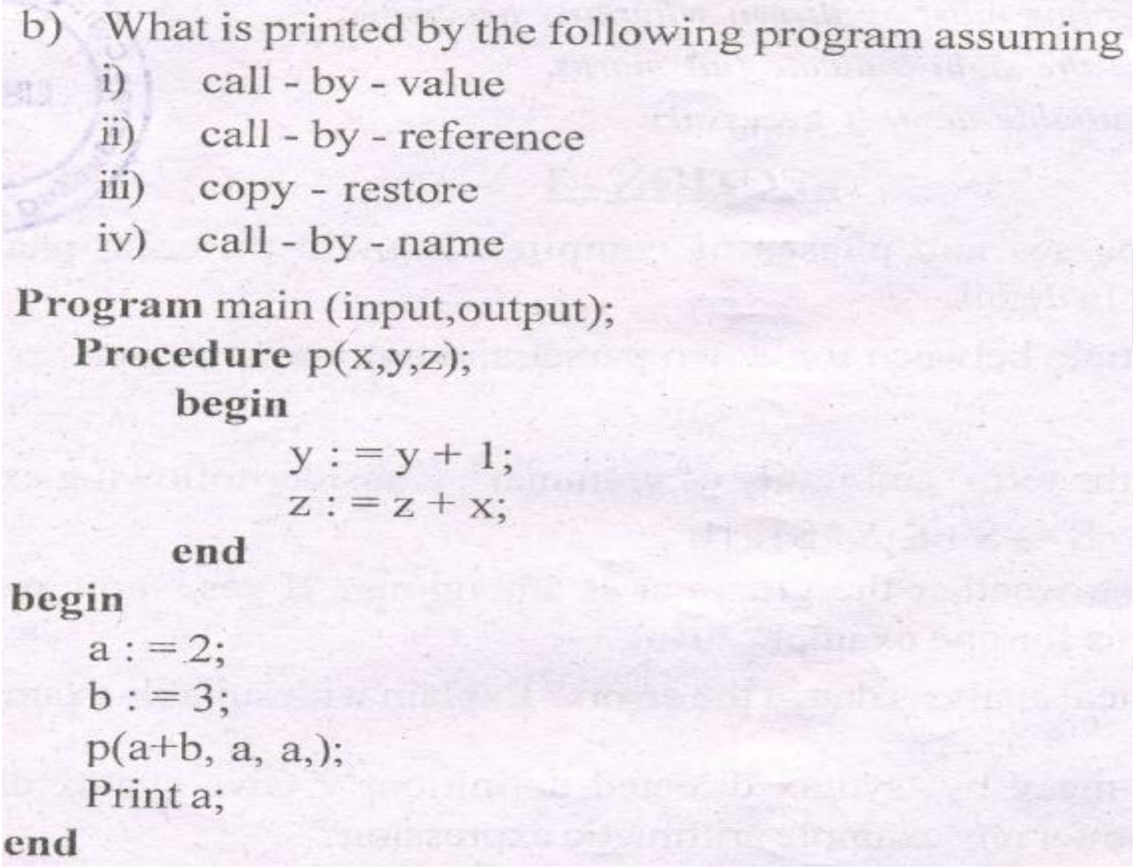
12.



Given following program. Show contents of activation record.

```
Procedure MAIN ( );
Procedure P(a);
Procedure Q(b);
L1 : R(x, y);
       end Q;
L2 :   Q(z);
       end P;
       Procedure R(c, d);
       end R;
L3 :   P(w);
L4 :   R(u, v);
       end MAIN;
L5 :   MAIN ( );
```

13.



b) What is printed by the following program assuming

i) call - by - value

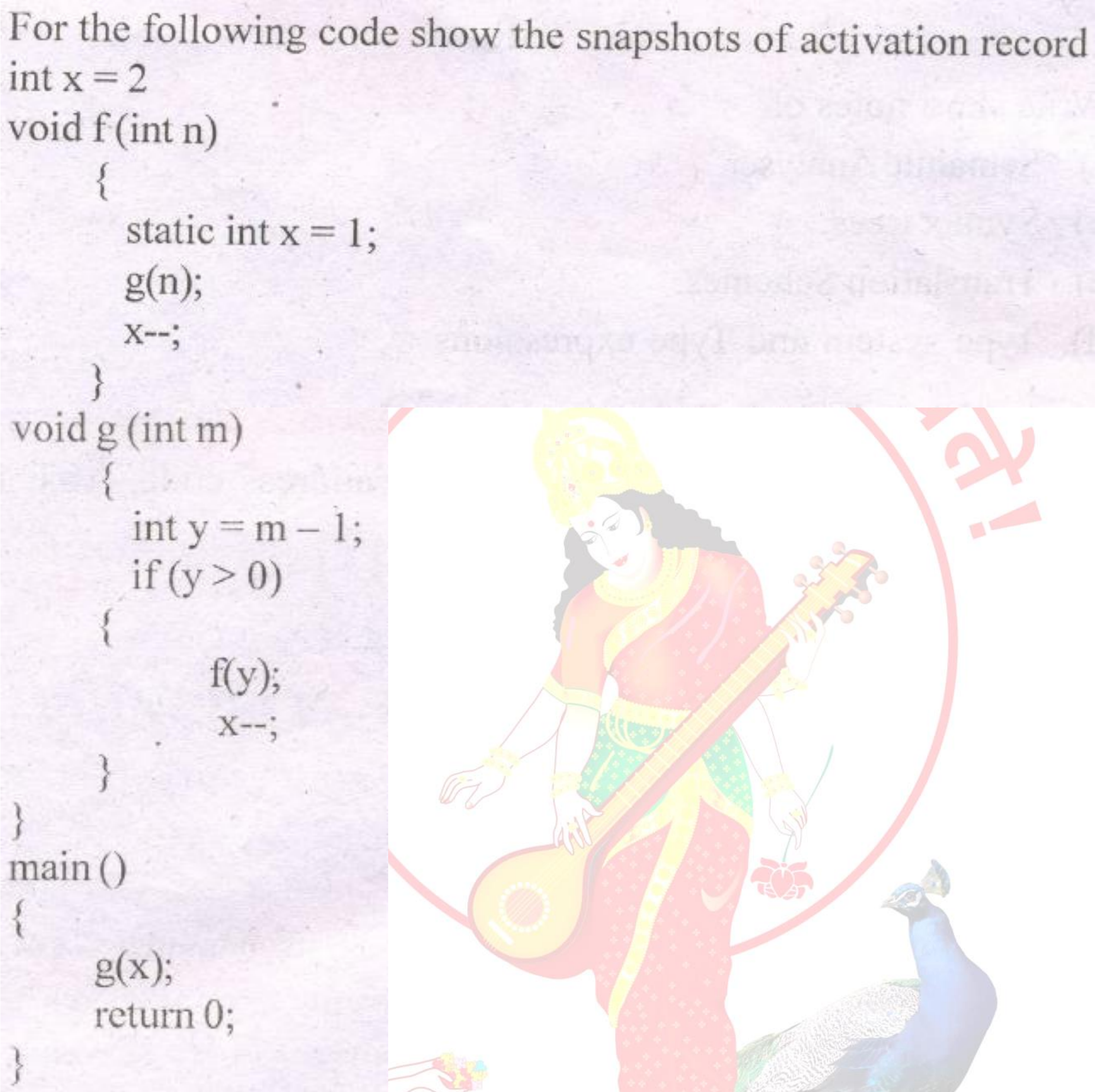
ii) call - by - reference

iii) copy - restore

iv) call - by - name

```
Program main (input,output);
   Procedure p(x,y,z);
        begin
              y : = y + 1;
              z : = z + x;
        end
begin
   a : = 2;
   b : = 3;
   p(a+b, a, a,);
   Print a;
end
```

14.



For the following code show the snapshots of activation record

```
int x = 2
void f (int n)
      {
         static int x = 1;
         g(n);
         x--;
      }
void g (int m)
      {
         int y = m – 1;
         if (y > 0)
      {
               f(y);
                x--;
      }
}
main ()
{
      g(x);
      return 0;
}
```

****************** **Best of Luck** *******************