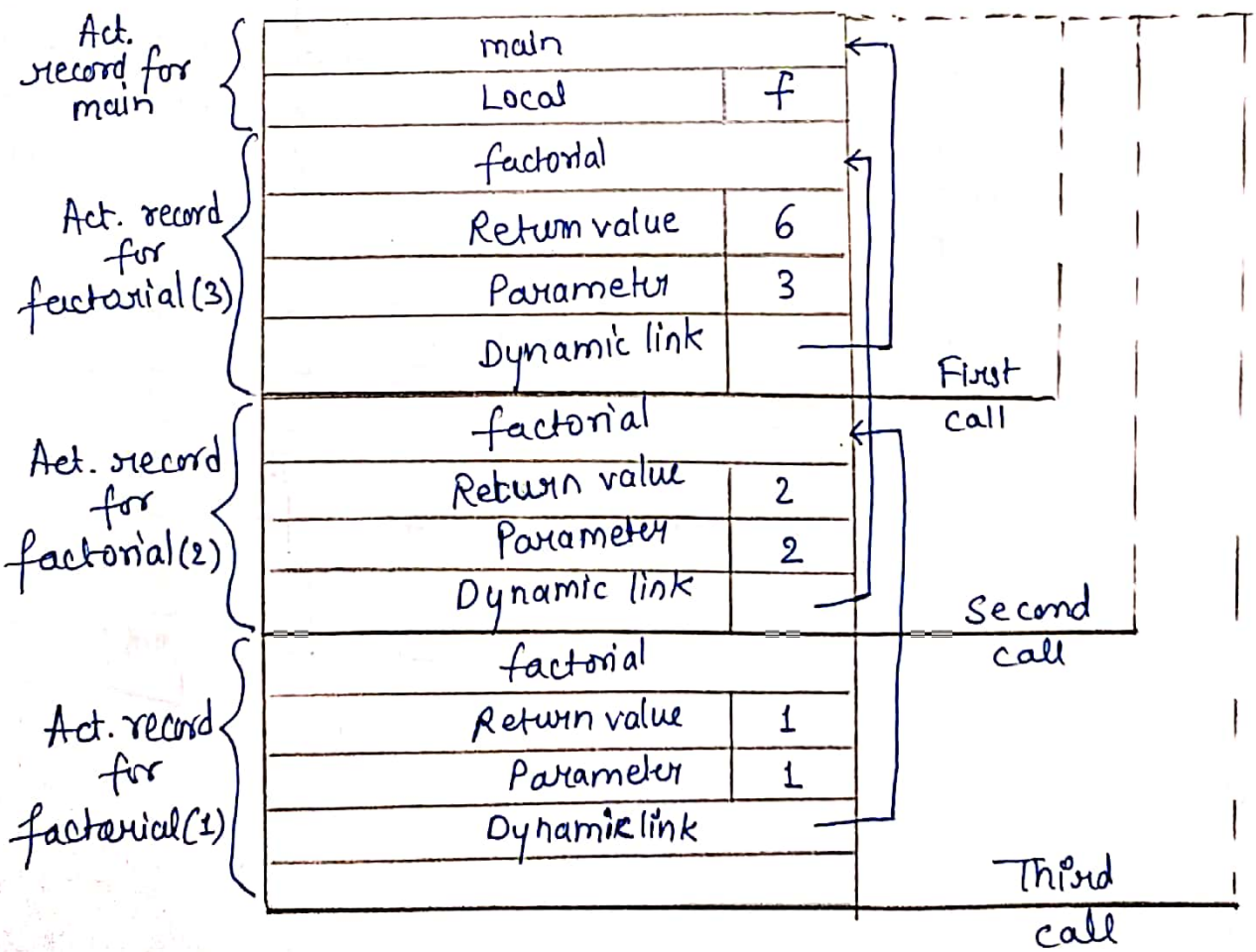
Unit - 4 (Compiler Design)

(Ques 1) By taking the example of factorial program explain how activation record will look like for every recursive call in case of factorial (3).

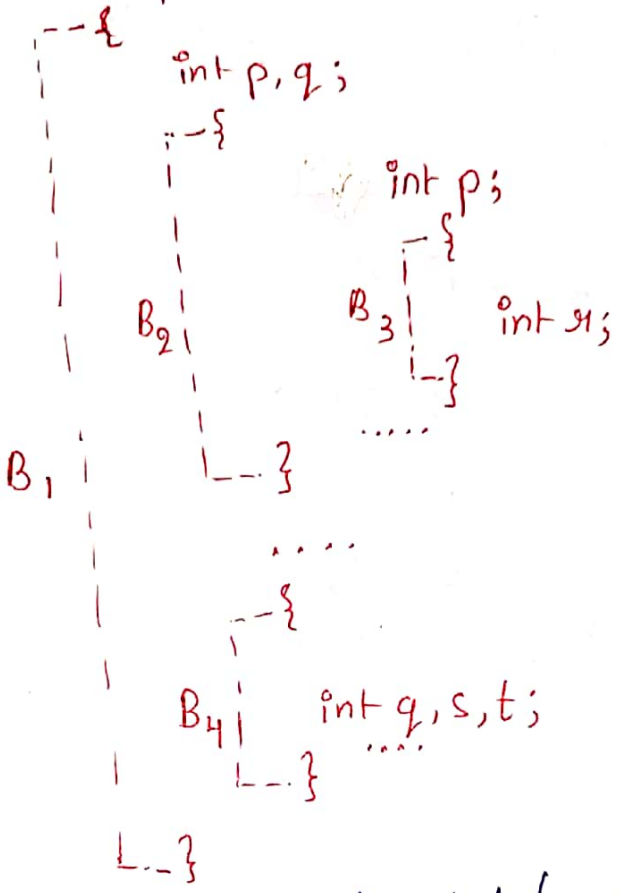
Solution:

```
main()
{
  int f;
  f = factorial(3)
}
int factorial(int n)
{
   if (n==1)
        return 1;
   else
        return (n*factorial(n-1));
}
```

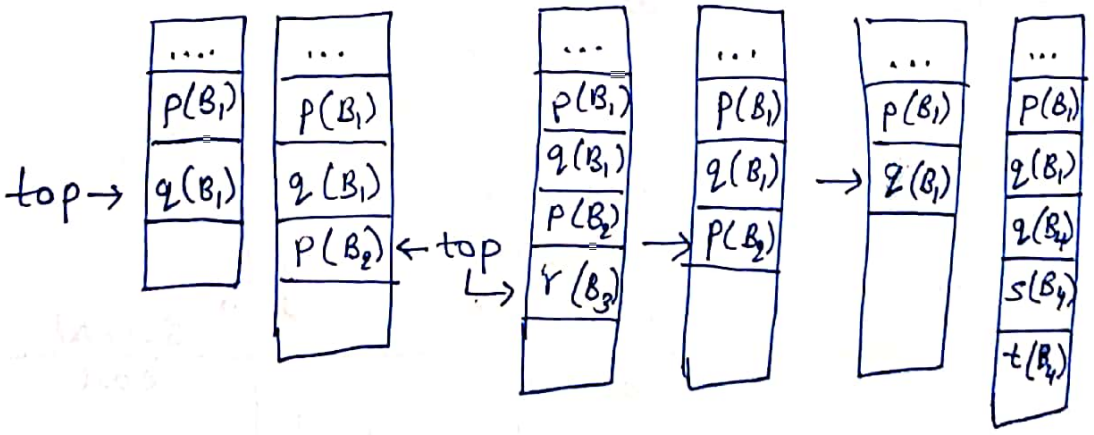


Ques 2) Obtain the static scope of the declarations made in the following piece of code.

```
scope_test()
{
    int p, q;
    {
        int p;
        {
            int r;
        }
    }
    {
        int q, s, t;
    }
}
```



Solution: The storage can be allocated for a complete procedure body at one time. The storage for the names corresponding to particular block can be as shown below:



Ques 3) What do you mean by Nested procedure?

Solution:- Nested procedure is a procedure that can be declared within another procedure. A procedure pi, can call any procedure that is its direct ancestor or older siblings of its direct ancestor.

Ques 4) What are the steps for calculating Nesting depth?

Solution:- Nesting depth of a procedure is used to implement lexical scope. The nesting depth can be calculated as follows:

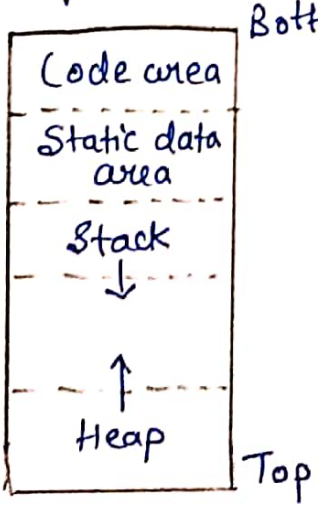
i) The nesting depth of main program is 1.
ii) Add 1 to depth each time when a new procedure begins
iii) Subtract 1 from depth each time when you exit from a nested procedure.
iv) The variable declared in specific procedure is associated with nesting depth.

Ques 5) What do you mean by Run time storage?

Solution:- The compiler demands for a block of memory to operating system. The compiler utilizes this block of memory for running (executing) the compiled program. This block of memory is called run time storage.

Ques 6) With the help of a diagram, show the subdivision of run-time memory which is allocated to compiler.

Solution:- The subdivision of run-time memory is shown by following figure -



Ques 7) What are the different storage allocation strategies?

Solution:- There are three different storage allocation strategies based on the division of run time storage. The strategies are -

1. Static allocation - The static allocation is for all data objects at compile time.
2. Stack allocation - In the stack allocation a stack is used to manage the run time storage.
3. Heap allocation - In heap allocation the heap is used to manage the dynamic memory allocation.

Ques 8) What are the limitations of static allocation & stack allocation?

Solution:- Limitations of static allocation

a) It can be done only if the size of data object is known at compile time.
b) The data structure cannot be created dynamically. That is, the static allocation cannot manage the allocation of memory at run time.
c) Recursive procedures are not supported by this type of allocation.

Limitations of Stack Allocation

- The memory addressing can be done using pointers and index registers. Hence this type of allocation is slower than static allocation.