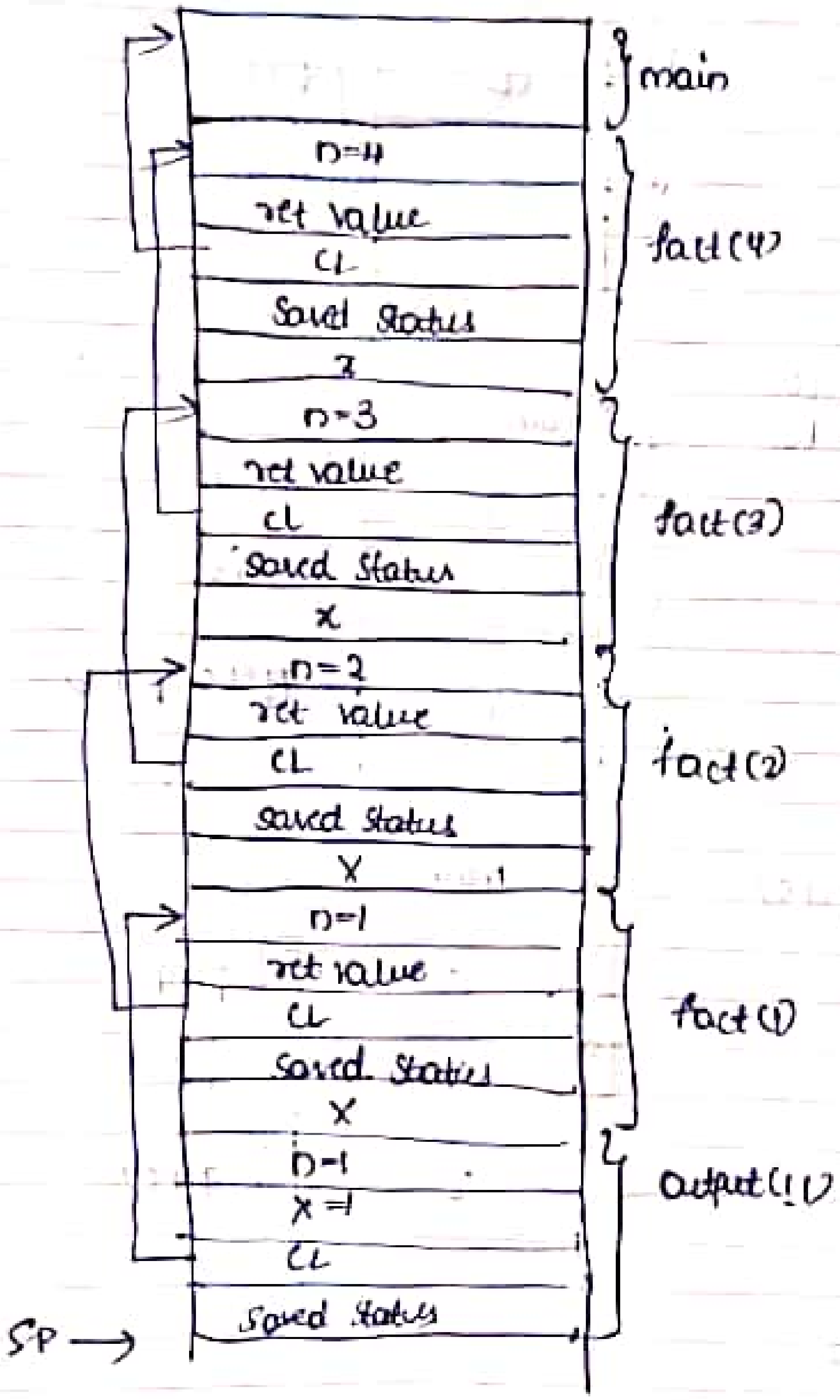
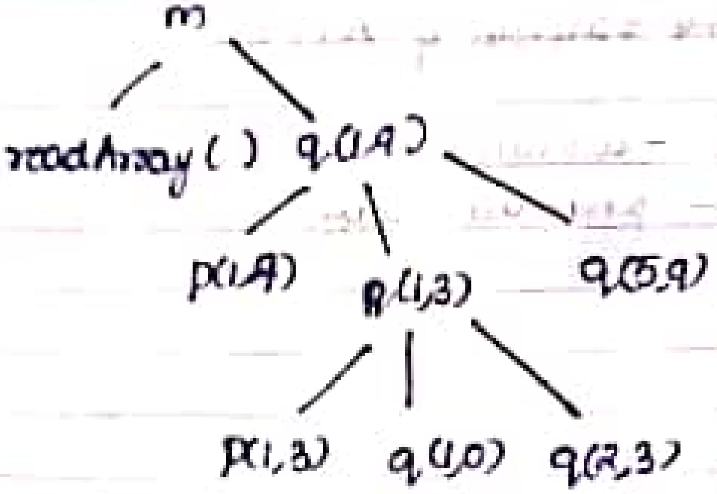
Activation Record Eg2 .

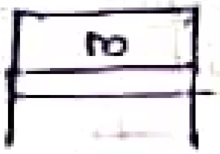


Activation Record - Eg: quicksort

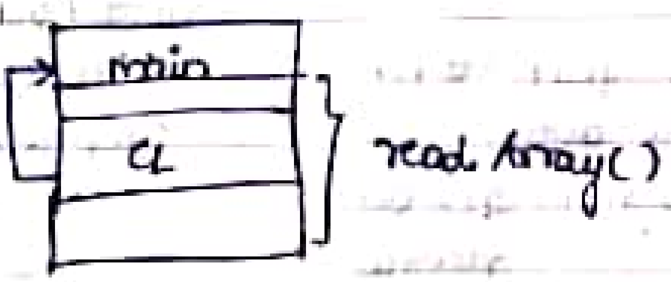
5



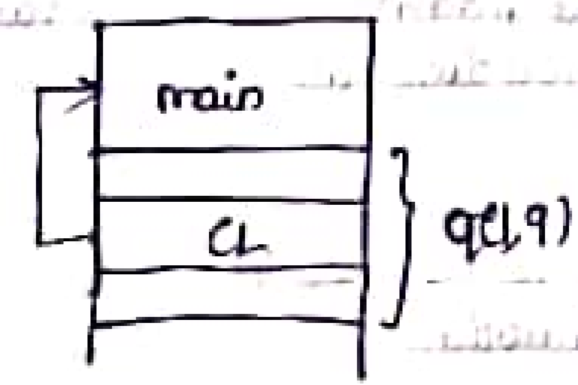
at m



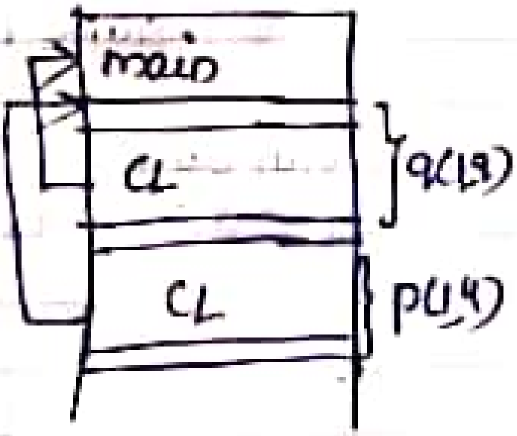
at read Array( )



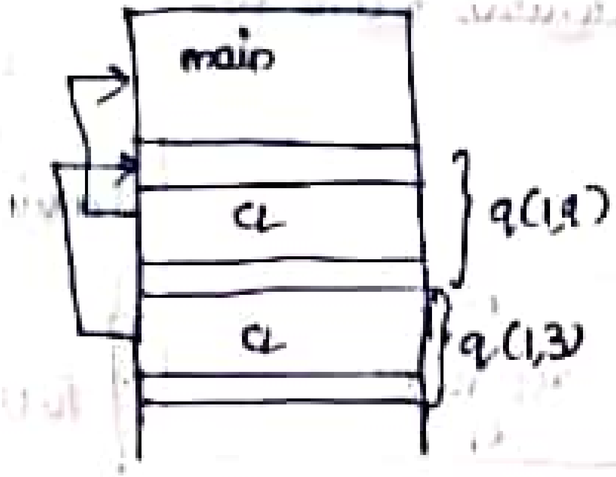
at q(1,9)



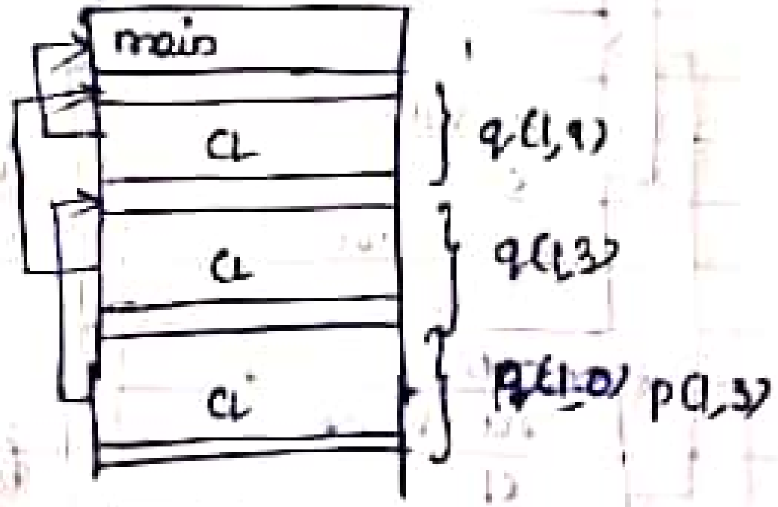
at p(1,4)



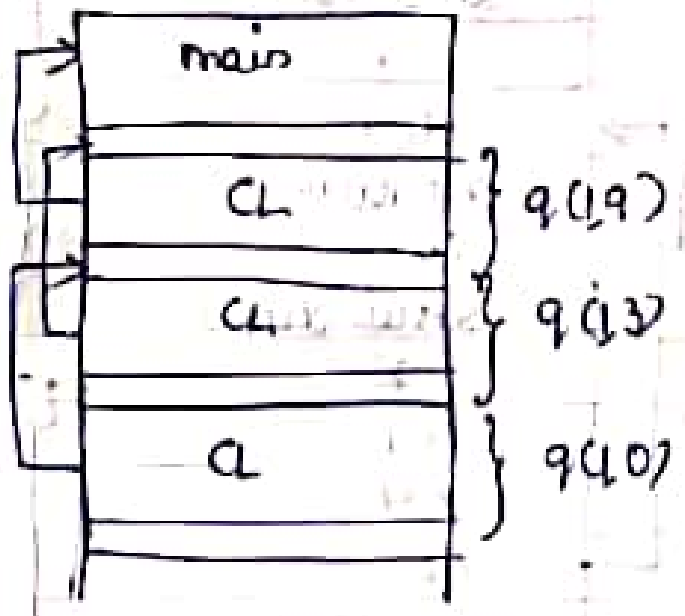
⑥ at q(1,3)



at ~~q(1,0)~~
p(1,3)



at q(1,0)



(4) Control Stack

→ Keep track of live procedure activation

→ flow of control in program corresponds to DF depth first traversal of Act. tree

push - activation begins
pop - when act. ends

Activation record

control link - pointer that points to activation record of caller

Actual Parameters
Return value
Control link
Access link
Saved Machine Status
Local data
Temporaries

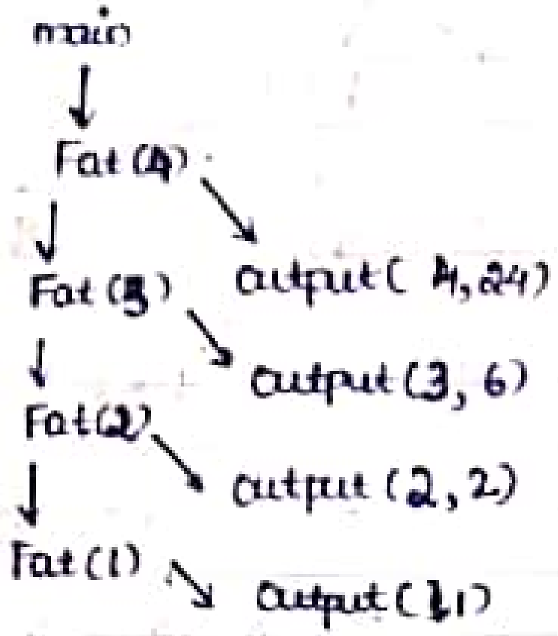
access link - points to non local data (maybe in other act. record)

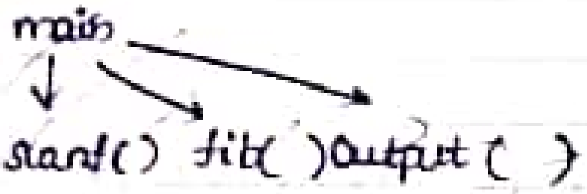
SMS - holds info about machine status before procedure call

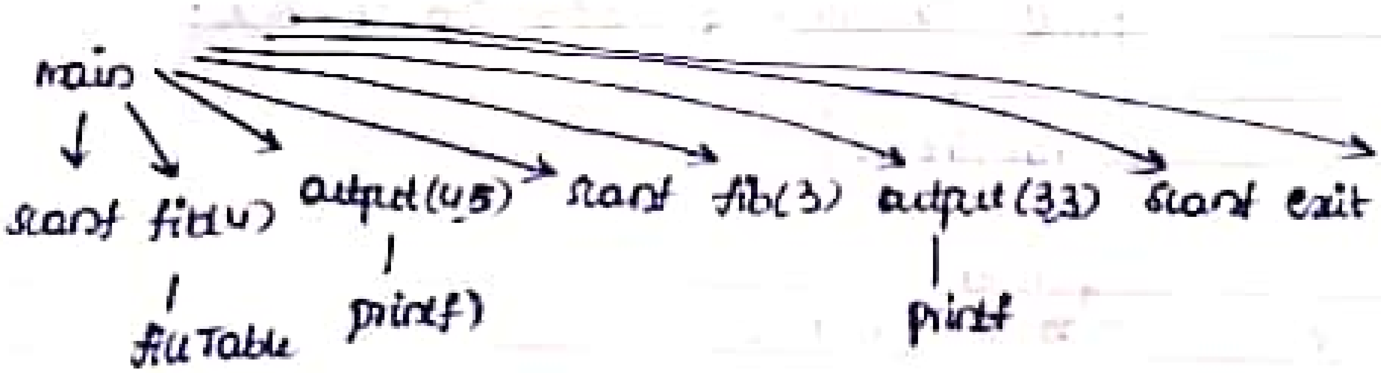
returned value - value to be returned

local data
temporaries - temporary values used in evaluation.

3 Activation tree Fat(4)







① Storage organisation

Procedure
- declaration that associates an identifier with a statement

Activation tree
Root - main
functions are children

Subdivision of Run time memory

compile time → Code → memory locations for code

compile time ↘ Static Data

Run time → Stack

~~Procedure~~

↓

Free Memory

↑

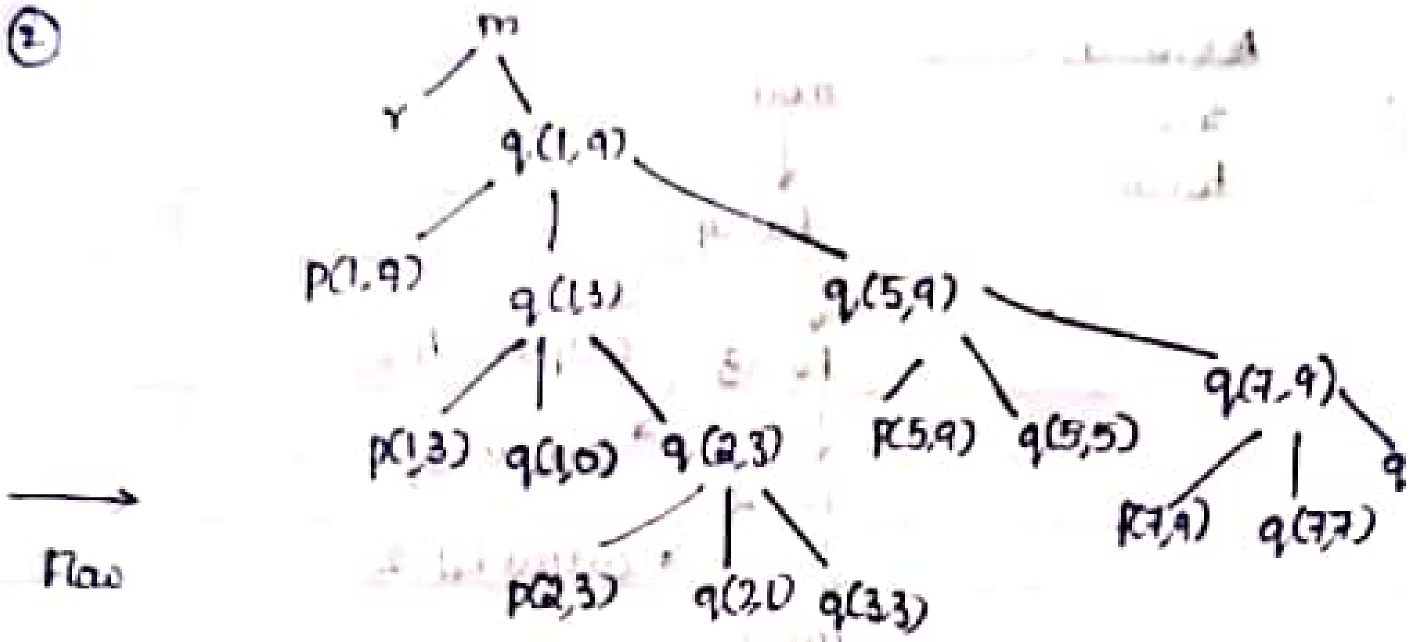
Run time Heap - ex. malloc

~~mai~~

```
enter main()
    enter readArray()
    leave readArray()
    enter qsort (1,9)
        enter part (1,9)
        leave part (1,9)

leave qsort (1,9)
leave main( )
```

②



Example - Quick Sort Activation tree

↳ helps us to know which all function are active and also the control flow

```
void output (int n, int x){
    printf ( "The value q %d! is %d.\n", n,x);
 ]
   int Fat (int n){
     int x ;
     if (n>1)
      x = n * Fat (n-1) ;
    else
       x=1;
    Output (n,x);
    return;
 }
 void main (){
     Fat (4);
 ]
```