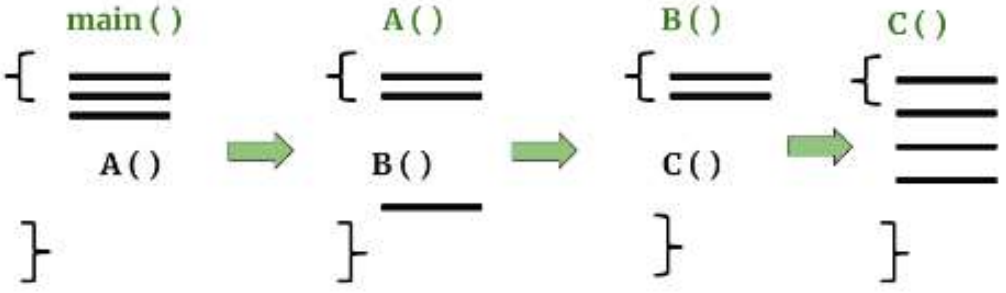
Control Stack/Run Time Stack

- Control stack / Run time stack – keeps track of live procedure activation
- A procedure name is pushed on to the stack when it is called (activation begins) and it is popped when it returns (activation ends)
- Information needed by a single execution of a procedure is managed using an activation record or frame.
- When a procedure is called, an activation record is pushed into the stack and as soon as the control returns to the caller function the activation record is popped.



Before going to A() activation record of main is created when A() is called activation record of A() is created.

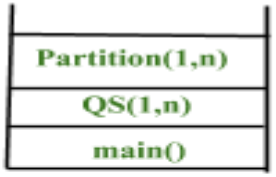
Before going to B() activation record of main and A() in stack.

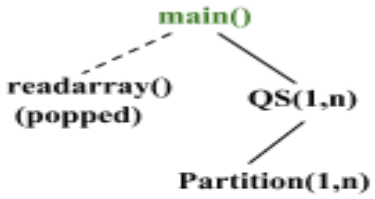
Before going to C() activation record of main and B() is pushed in stack.

Control Stack/Run Time Stack

readarray() execution completes (popped from stack).

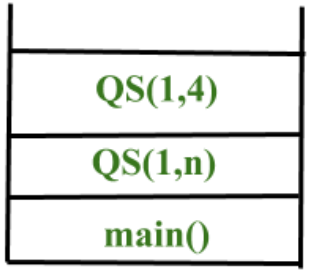
QS is called so it Enters the Stack.

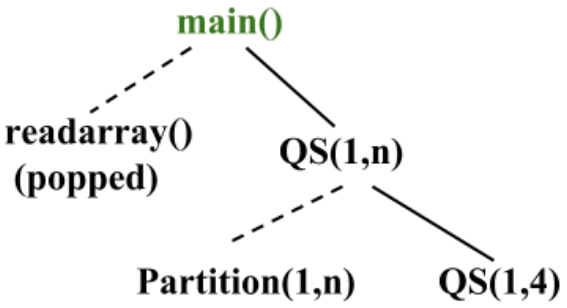




Partition Execution completed (popped out of stack)

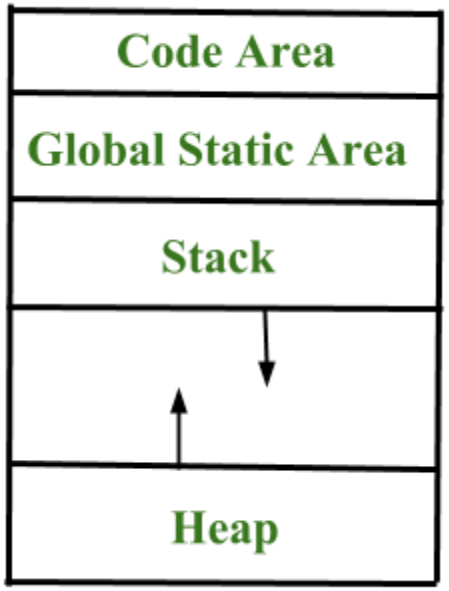
Now QS is called again so it enters the Stack.





Storage Organization

- Runtime storage can be sub divided to hold
  - Target Code – Text part (Memory requirement is known at compile time) – doesn't change @runtime
  - Static data objects
  - Automatic data objects is Stack – procedure call random manner *[stack – procedure]*
  - *Dynamic Data Objects is Heap* – managing memory allocation of memory for *variables* @ runtime



The Stack grows towards Higher Memory.

Heap grows towards lower Memory.

Storage Allocation Strategies

- *Static Storage Allocation*
  - Recursion is not supported
  - lays out storage for all data objects at compile time
  - Ex: FORTRAN
- *Stack Storage Allocation*
  - Recursion is supported
  - manages the run-time storage as a stack. (procedure is called its activation is pushed into stack)
  - Stack activation are pushed and popped
- *Heap Storage Allocation*
  - Recursion is supported
  - allocates and deallocates storage as needed at run time from a data area known as heap. Example:malloc