

2) Here are the 4 pairs of new* and undo* routines (they should have done similar operations):

```
var oldseek: LOCATION;  
var oldbuf: UNIT;
```

newseek(j)

undoseek(j)

```
oldseek := i; (* keep current pointer )  
seek(j)
```

```
seek(oldseek); (* restore previous pointer *)
```

buf := newfetch()

buf := undofetch()

```
oldbuf := buf; (* keep old buffer )  
return(fetch());
```

```
return(oldbuf); (* restore old buffer *)
```

newinsert(buf)

undoinsert()

```
insert(buf);
```

```
delete()
```

newdelete()

undodelete()

```
oldbuf := buf;  
buf := fetch();  
delete();
```

```
insert(buf);  
buf := oldbuf;
```

nothing special to discuss here really but they might have talked about it ☺

3) lseek is ready to move the current pointer to a $j > n$ (after the EOF of the file) and then you can write from that j onwards (assume this was done so there is a hole before). If an attempt to read from position $n \rightarrow j-1$ is made, a zero is returned. So to model it in the 'Stream Model', holeseek just inserts $j-n$ units with the value zero in them (that should be interpreted as a zero of lseek) and leaves the current pointer at position j as it should be.

holeseek(j)

```
if j <= n then seek (j);  
else begin  
  buf = 0;  
  seek(n);
```

```
m := i;
for k := m to j-1 do begin
    insert(buf);
    seek(i+1);
end (* do begin *)
end (* begin *)
```

4) On a Tape, a Physical Record was simply a collection of Logical Records for a length of a few (tens of) thousands of bytes, i.e., a Physical Record was what we call now (in the Units Model/Hierarchy) a Block. (Note: This is a simple question/answer – whoever gives more unneeded details or got confused with it – it's his/her problem 😊).