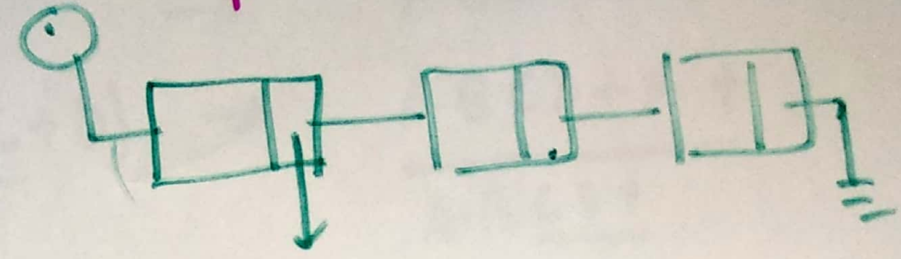
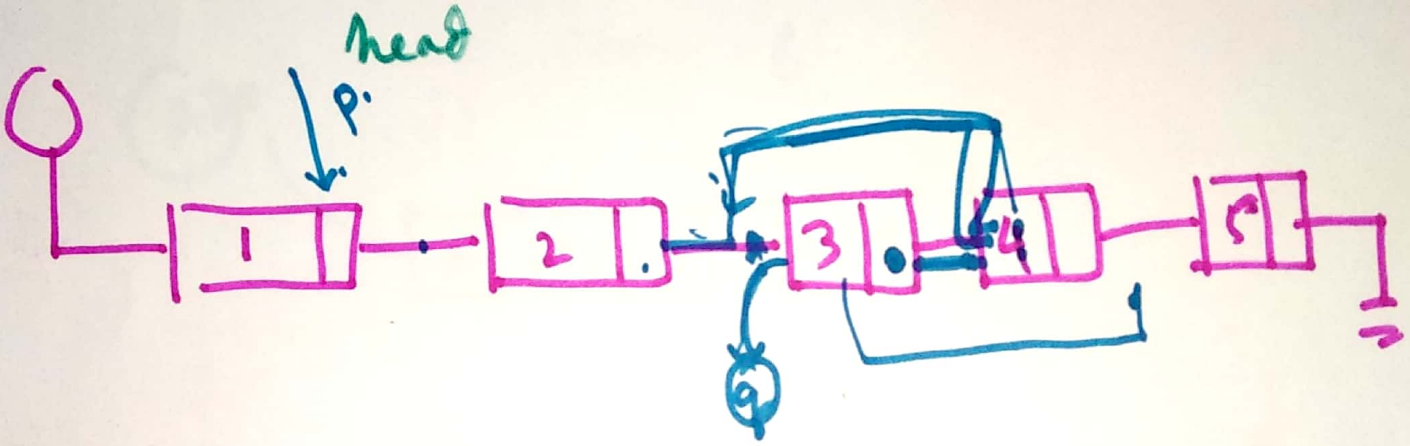
link list

9/01/18.

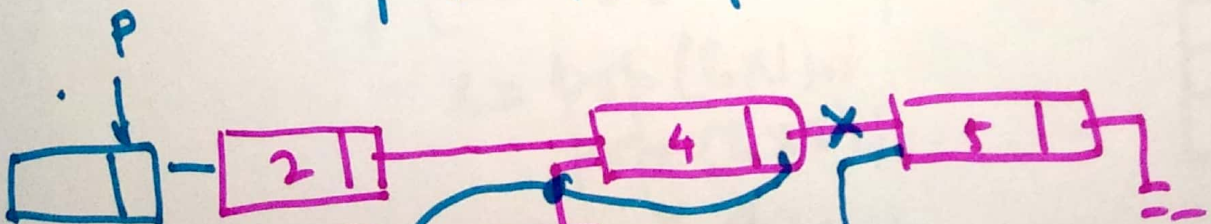
```
{
struct node *next;
}
```



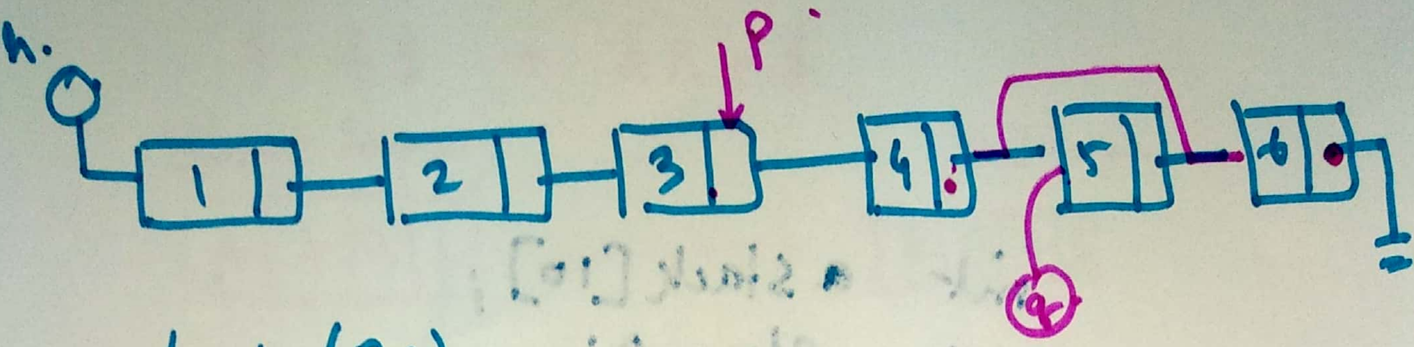


p → next. =
p → next → next.

q = p → next;
p → next = p → next → next.



q → next = p → next → next;
p → next → next = q;



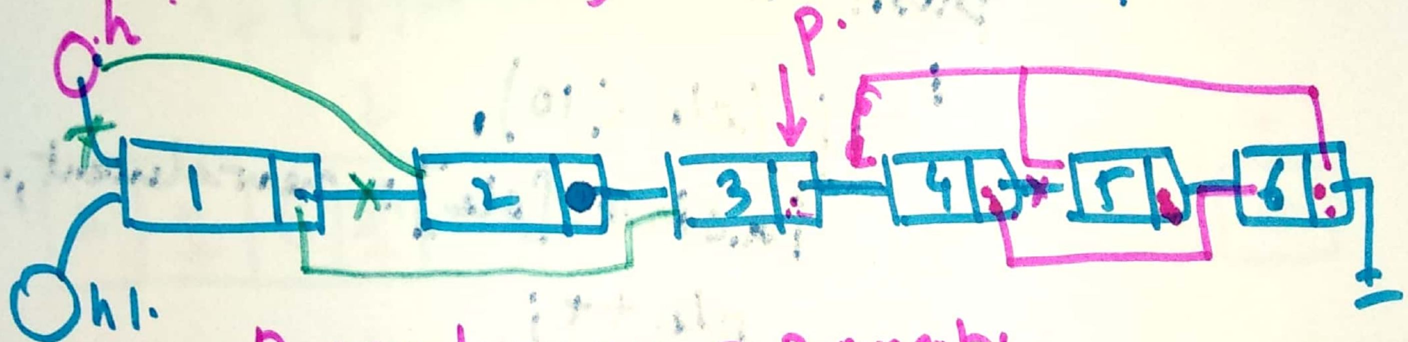
swap (5,6).

q = p → next;

p → next = p → next → next;

p → next → next = q;

/ p → next → next → next or q → next = NULL;



p → next → n → n = p → next;

p → next = p → next → next;

p → next → n → n = NULL;

→ change (1 & 2) swap (1,2).

h1 = h;

h = h → next;

h1 → next = h → next;

h → next = h1;

```
int  astack[10];
int  ele = 0;
int  max_ele = 10;
```

```
/* Global variable */

~~push (ele)~~.
push (int newelement) / int *stack )
  {
    if (ele < 10),
     {astack[ele] = newelement;
      ele ++;
     } return (1);.
    }
   return (0);
  }

pop ( ), int *popelement);
 { if (ele != 0)
    { *popelement = stack[ele];
       ele --;
      return (1);
 }  }
   return (0).
 }
```

LIFO

Last In First out

infix Postfix.

A + B $\rightarrow$ AB+.

A+B*C $\rightarrow$ ABC*+ .

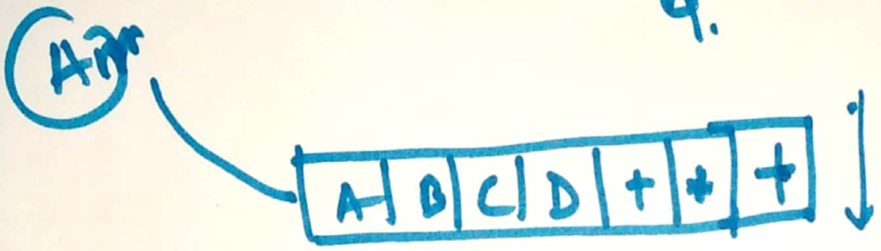
A+B*(C+D) $\rightarrow$ ABCD+*+

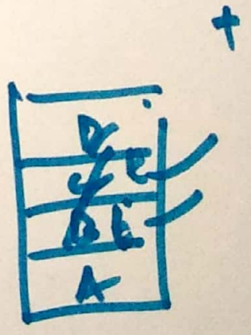
$\overline{\text{AB}\underline{\text{CD+}}}$

A $\underline{\text{BE} \quad *}$

A F +

G.

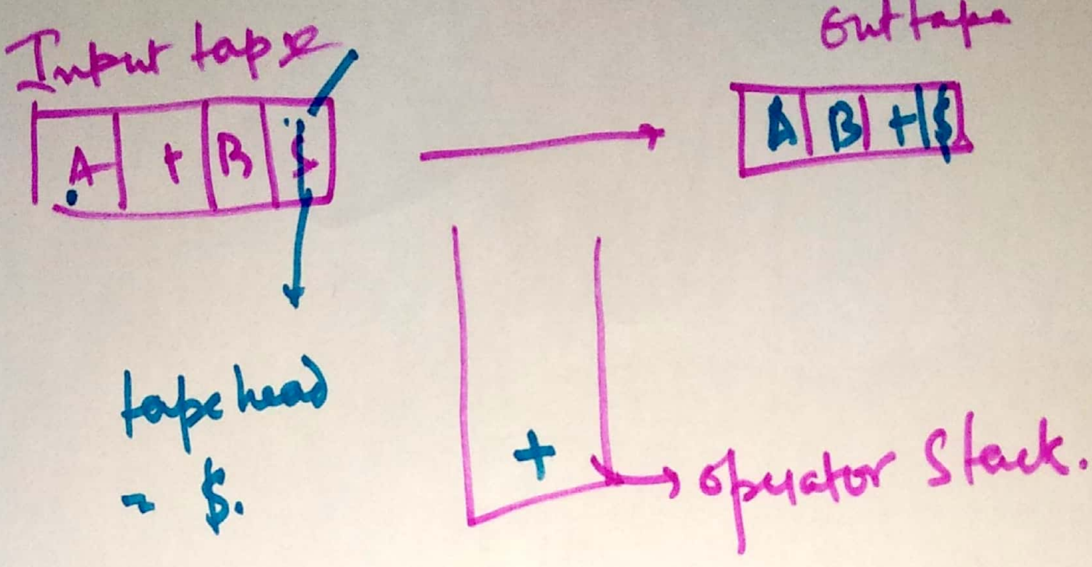




```
for(i=0;i<n;i++).
{
  if (ARR[i] == operand).
      push(ARR[i]);

   if (ARR[i] == operator)
      x = pop(&x1);
          pop(&x2);
          c = x1 ⊕ x2;
          push(c);
}.
```

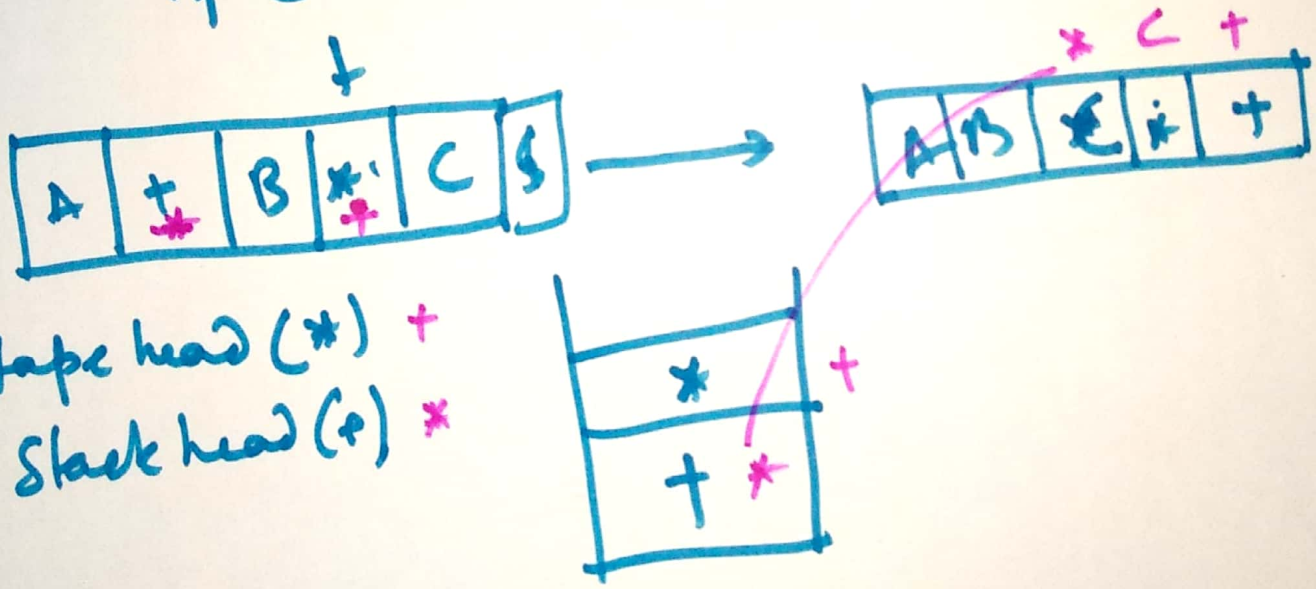
$A + B\,\$ \rightarrow AB + \$$



tape head = $.

Stack.head +

if (.$) at tape head flush the stack.



tape head (*) +

Stack head (+) *

A + (B * C)