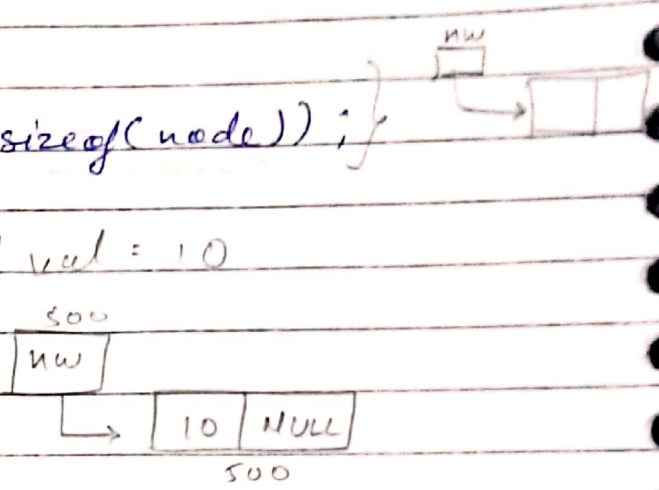
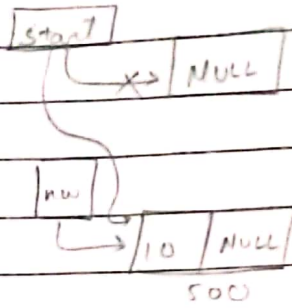
LINKED LIST

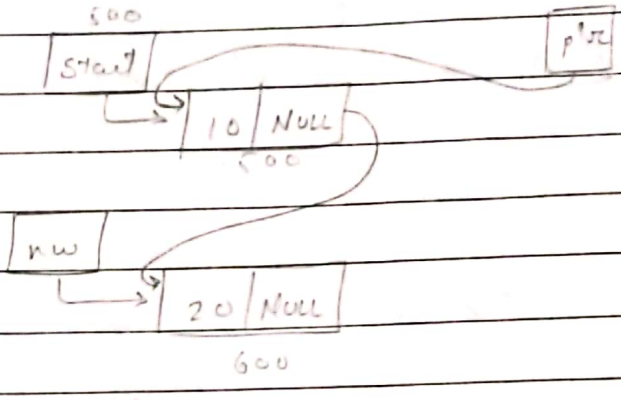
```
void insert ()
{ int val;
  nw = (node *) malloc (sizeof(node));
  pf ("Enter Value : ");
  sf ("%d", &val);   // val = 10

  nw -> info = val;
  nw -> link = NULL;
```

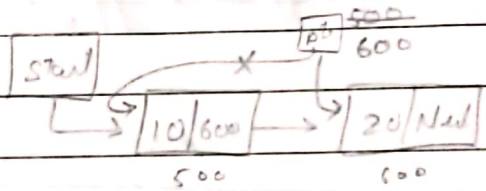



```
  while ( ptr -> link != NULL )
  {
    ptr = ptr -> link;
  }
  ptr -> link = nw;
}

if ( start == NULL )
{
  start = nw;
}
else
{ ptr = start;
  while ( ptr -> link != NULL )
  {
    ptr = ptr -> link;
  }
  ptr -> link = nw;
}
}
```

Teacher's Signature.....








```
void display()
{ ptr = start;
  printf("START --> ");
  while ( ptr != NULL)
  {
    printf("%d -->", ptr -> info);
    ptr = ptr -> link;
  }
printf("NULL");
}
```

START --> 10 --> 20 --> NULL

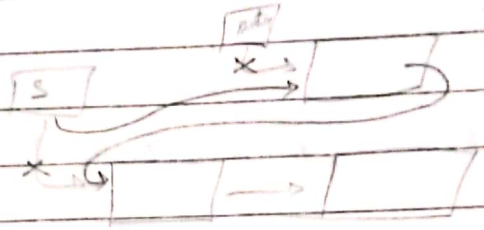

```
void delete_at_first()
{
    ptr = start;
    start = ptr -> link;
    ptr -> link = NULL;

    printf("Deleted value : ", ptr -> info);
}
```

```
void delete_at_last()
{ node *ptr2 (=NULL), *ptr1 (=NULL);
    ptr1 = start;
    while(ptr1 -> link != NULL)
    {
        ptr2 = ptr1;
        ptr1 = ptr1 -> link;
    }
    ptr2 -> link = NULL;

    printf("Deleted value :
}
```

Teacher's Signature.....

```
void delete_at_value()
{
    node *ptr2, *ptr1;
    int flag = 0;
    ptr2 = NULL;
    ptr1 = start;

    while ( ptr1 != NULL)
    {
        if ( ptr1 -> info == val)
        {
            flag = 1; break;
        }
        else
        {
            ptr2 = ptr1;
            ptr1 = ptr1 -> link;
        }
    }
    if ( flag == 1)
    {
        ptr2 -> link = ptr1 -> link;
        ptr1 -> link = NULL;
        printf (" Deleted value :", ptr1 -> info );
    }
    else
    {
        printf (" Not found, Deleted not possible ");
    }
}
```

```
void insert_at_begin()
{
    int val;
    ptr = (node *) malloc (sizeof(node));
    p.f("Enter Value:");
    s.f("%d", &val);

    ptr->info = val;
    ptr->link = NULL;

    ptr->link = start;
    start = ptr;
    free(ptr);
}
```

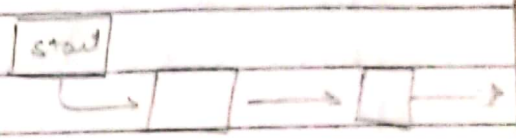



```
void insert_at_value()
{ int val, flag = 0, item;
    ptr = (node *) malloc (sizeof(node));
    p.f("Enter Value:");
    s.f("%d", &val);

    new ptr -> info = val;
    new ptr -> link = NULL;

    p.f("Enter value after which you want to enter:");
    s.f("%d", &item);
```

Continue →

Teacher's Signature

```
ptr = start;
while ( ptr != NULL)
{
   if ( ptr -> info == item)
   {
      flag = 1 ; break;
   }
   else
   {
      ptr = ptr -> link;
   }
}
   if ( flag == 1)
   {
      new -> link = ptr -> link;
      ptr -> link = new;
      free ( ptr);
      free ( new);
   }
   else
   {
      printf (" Not found, Insertion not possible");
   }
}
```



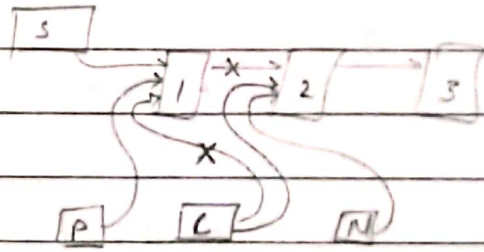
Teacher's Signature.....

```
void search()
{ int flag = 0; val, pos = 0;
  ptr = start;
  while ( ptr != NULL)
  {
    if ( ptr
```

```
  pf (" Enter Search value : ");
  sf ("%d", &val);

  while ( ptr != NULL)
  { pos ++;
    if ( ptr → info == val)
    {
      flag = 1 ; break;
    }
    else
    {
      ptr = ptr → link;
    }
  }
  if ( flag == 1)
  {
    pf (" Value found at position %d ", pos);
  }
  else
  {
    pf (" Value not found ");
  }
}
```

Teacher's Signature.....

```
void sort()
{  node *i, *j; int temp;

   for (i = start; i != NULL; i = i->link)
   {
      for (j = i->link; j != NULL; j = j->link)
      {
         if (i->info > j->info)
         {
            temp = i->info;
            i->info = j->info;
            j->info = temp;
         }
      }
   }
}
void reverse()
{  node *prev = NULL, *next = NULL, *curr;
   curr = start;

   while (curr != NULL)
   {
      next = curr->link;
      curr->link = prev;
      prev = curr;
      curr = next;
   }
   start = prev;
}
```



THEORY

STACK

Expressions Types :- infix → A + B
postfix → AB +
prefix → + AB

Priority :-

High	( )
↕	^
	* /
low	+ -

→ Associativity

A * (B+D) / E - F * (G + H / K)

→ A * (B+D) / E - F * (G + HK/)

→ A * (BD+) / E - F * (GHK/+)

→ A * /BD+/ E - F * GHK/+

→ ABD+* / E - FGHK/+*

→ ABD+*E/ - FGHK/+*

→ ABD+*E/FGHK/+*-

Teacher's Signature.....

Step 1 :- (A * (B+D)/E - F * (G+H/K))

Symbol	Stack	Postfix
(	(	
A		A
*	(*	A
(	(* (	A
B	(* (	AB
+	(* (+	AB
D	(* (+	ABD
)	(*	ABD+
/	(* /	ABD+*
E	(/	ABD+*E
-	(-	ABD+*E/
F	(-	ABD+*E/F
*	(- *	ABD+*E/F
(	(- * (	"
G	(- * (	ABD+*E/FG
+	(- * (+	"
H	(- * (+	ABD+*E/FGH
/	(- * (+ /	ABD+*E/FGH
K	(- * (+ /	ABD+*E/FGHK
)	(- *	ABD+*E/FGHK/+
)	-	ABD+*E/FGHK/+*-

Teacher's Signature.....

Evaluate : 12, 7, 3, -, 1, 2, 1, 5, +, *, +

~~Symb~~

12, 7, 3, -, 1, 2, 1, 5, +, *, +)

Symbol	Result
12	12
7	12 7
3	12 7 3
-	12 4
1	3
2	3 2
1	3 2 1
5	3 2 1 5
+	3 2 6
*	3 12
+	15
)	Stop

Ans = 15

Teacher's Signature.....