```
Global Structure:
Doubly dinked dist
                               Struct node *create-11 (struct node *start)
       Street node * new node, *ptr;
        int num;
        printf ("In Enter -1 to end");
        printf ("In Pukr thedata: ");
        scanf ("%d", & num);
        while (num!=-1)
           १
               if ( start == NULL)
                   new_node = (storet node*) malloe (sizeof (struct node));
                   new-node -> prev= NULK) ( oct ) [ d - torace & day
                   mew-node -> data = mum;
                    new-node -> next = NULL) 16 mm 1 16 mm
                     start = new_node;
                                       intelled the start of
                   3
                 esse
                           intelled the day of the thirty
                     ptr = start;
                     new-node = (struct node *) malloc (site of (struct node));
                      new-node -> data = num;
                      while (ptr->next != NULL)
                              ptr=ptr-next;
                          ptr -> next = new-node;
                           new-node -> prev= ptr)
                            new-node -> next= NULL;
                         printy ("In Enter the data:");
                          scanf ("%d", 6 rum);
                        return start;
                    3
```

```
struct node * display (struct node *start)
       struct mode *ptr;
         ptr = start;
        while (ptrl=NULL)
Struct node * insept_beg (struct node *start)
      struct mode * new-node;
       int num;
       printf ("14 futer the data: ");
       scanf ("%d", & num);
      new-node = (struct mode *) malloc (sire of (struct node));
       new-node -> data = num;
       start -> prev = new_node;
       mew-node -> next = start;
       new-node -> prey = NULL; /2/22/19 19
         start = new-node;
         return start;
```

```
struct node * insert_end (struct node * stort)
      struct mode * ptr, * new-node;
       int mum;
       print ("In their the data:");
                                          Windshid work & M" I forty
       scanf ("%d", Onum);
      new-yode = (struct node *) malloc (size of (struct node));
                                  14 style solor 146 1167)
        new-node -> data = num;
         ptr = start;
         while (ptr-mext! = wold) 11) 1011-17 (1111-1111)
               per= per -> next;
            ptr + next = new.node;
            new-node + prev= ptr;
             mew-node -> next = NULL;
             return (Hart;
        3
                                       illy make shouldow
                                : from 2- of i from 4- of the ha
Struct node * insert-before (struct node * start) = 1000 00119
                                  illian and a Penn
       Struct node * new-node, *ptr;
                                          Charles months
         int mum, val;
        printfi"In Enter the data ");
        sconf ("%d", wnum);
         printf ("In Enter the value before which the data has to be
                    inserted");
         scanf ("%d", wal);
          new-node = (struct node*) mallor (size of (struct node));
          new-node -> data = num;
                                        211711 VIII
           ptr= start)
           while (ptr + data ! = ral)
             new-node -> next = ptr)
              new-node -> prev = ptr -> prev;
               ptr -> prev -> next = new-node;
               ptr -> prev= new-node)
               return start)
```

```
I sell to from a stone tourts
Struct node * insert-after ( struct node * start)
   struct node * new-node, * pt Y;
    int num, val;
                                                -16 art 1) w/ 1 (strict
    printf ("In Euter the data:");
                                                ( much. "to?") from
    sconf ("0/0d", 6 num);
    printf ("In Enter the value after which the data has to be incerted: ");
    8 conf ( "%d", & val);
      new-node = (struct node*) malloc (site of (struct node));
       new-node -> data= num;
                                             : 1211 2.27 111
         ptr = start)
              ptr=ptr->next;
                                                 itoute multir
             new-node ->prev=ptr)
              new-node > next = ptr > next;
               ptr > next > previa new mode; 12) orapid . horris & above to 12
                 ptr > next = new-node;
                                       irly & , Shar -once & other trusts
                  return stant;
                                                    (LOV, WALL) Fr
                                       d" which with a tring of " I for "
                                            ( ( minc) , " boy") 100
   struct node *delete-beg (struct node *start)
         Ž
                                            · ( " halos 11
            smuct made *ptr;
              ptr= start;
              Start = start -> next;
              stait -> prev = NULL;
              free(ptr);
              return start;
```

```
struct mode * delete_end ( struct node * start)
      smuct mode *ptr)
       while (ptr->next != NULL)
              ptr=ptr > next;
           free (ptr);
struct node * delete-after (struct node * stant)
        shuet node *ptr, *temp;
        int val;
      printf ("In take the value afterwhich the node has to eleleted:");
        scomf (" god", wral);
         ptr = start;
         while (ptr->data!=val)
                 ptr = ptr -> next;
            temp = ptr-next;
             free (tump))
              return start;
```

```
struct node * delete-before (struct node * start)
                                                  Mr. Mild, & Main Mills
   struct node *ptr, *temp;
                                                 - It I'm where I water
   int vals
    printf ("In fater the value before which the node has to be deleted!");
    Acanf ("%d", 6 val);
      ptr = start)
     while (ptr > data != val)
                         the prev=ptr; if (temp==stant)

the prev=ptr; stant = delete-beg(stant);
  Struct mode a delête-lut (struct node *stast)
            while (start != NULL) say a regression
                       start = delete-beg (start);
                  return start;
```