new.node\_) next = NULL; new.node\_) rea = NULL;

92 Phr = head copile (ptr-> nex != NULL) Now- nog en-) then = bel! new-node-s mext ł } } void insert-left() struct node " new-rode o; \* pte; (int val, num; new-node = (Street node \*) molloc (size of (Struck-node)); Print f ("enter a value to injertal lett:11); scanf 1". (d", & val); printf (11 Enter the value of node: 11); Scans "yd", Unum! new-node - data = val; Phr=head; while of phr->data != num) 4 by = bpc -> vexti Ptr-) prev-) next = nev-nod () new-node -sprew - ptr-sprev; non-words - west = bfx; bee spron = non-node; ¥

```
void diplay ()
      if ( head & = NUL2)
            bring the Tril outty");
          while (ptr-I next = NULL).
                exprint f ("1. 1. 13", ptroducta);
                     Ptr = Dtr -> neart;
void del ()
4
     Smeet Smuch hode "phi;
      int val;
      printf!" enter the value table deleted");
       scart ("1.d" [ wol);
        pto = head;
         ort (head solute z val):
                beag = they want;
           Use
              while ( oh - & data ), = val)
                  pti= pti-snock;
                 bu-) bon y west = bud west;
                 berspect & ben = ben ben;
          6 free (ptr);
```

olpi 1. Oveale-11 d. mant a left 3-delete 4. d'splay S. exit \ . . Enter -180 exit: Enhu the raim! 32 Enter the num: "H6" Enter the num: 61' Enter the num : - 1 39->46-765. 2 enter a value to intert at 1 cft: 39 enter the nat of nade: Hb 4 32799746765 93 conter the value to be deleted : x16 4 32-399-365

```
#include <stdio.h>
1
2
     #include <stdlib.h>
3
4
     struct node
5
6
         int data;
7
         struct node *next;
8
         struct node *prev:
9
     };
10
11
     struct node *head;
12
13
     void create 11()
     1
14
15
         struct node *new_node, *ptr;
16
         int num:
17
         printf("Enter -1 to exit.. \n");
         // printf("Enter the num: ");
18
         // scanf("%d", &num);
19
         while (num != -1)
20
21
         1
22
              printf("Enter the num: ");
             scanf("%d", &num);
23
              new_node = (struct node *)malloc(sizeof(struct node));
24
25
              new_node->data = num;
              if (head -- NULL)
26
27
                  head - new_node;
28
29
                  new_node->next = NULL;
30
                  new_node->prev = NULL;
31
              else
32
33
                  ptr = head;
34
                  while (ptr->next != NULL)
35
36
37
                      ptr = ptr->next;
38
                 ptr->next = new_node;
39
                 new_node->prev = ptr;
40
                 new_node->next = NULL;
41
42
43
44
45
```

```
void insert_left()
    struct node *new node, *ptr;
    int val, num;
    new_node = (struct node *)malloc(sizeof(struct node));
    printf("enter a value to insert at left:");
    scanf("%d", &val);
    printf("Enter the value of node:");
    scanf("%d", &num);
    new_node->data = val;
    ptr = head;
    if (head == NULL)
        printf("list is empty!");
    else
        while (ptr->data != num)
            ptr = ptr->next;
        ptr->prev->next = new_node;
        new_node->prev = ptr->prev;
        new_node->next = ptr;
        ptr->prev = new_node;
```

```
struct node *ptr;
    if (head == NULL)
        printf("Linked list is empty!");
    else
        ptr = head;
        while (ptr->next != NULL)
        €
            printf("%d->", ptr->data);
            ptr = ptr->next;
void del()
    struct node *ptr;
    int val:
    printf("enter the value to be deleted:");
    scanf("%d", &val);
   ptr = head;
   if (head->data == val)
        ptr = ptr->next;
       head = ptr;
   else
   {
       while (ptr->data != val)
           ptr = ptr->next;
       ptr->prev->next = ptr->next;
       ptr->next->prev = ptr->prev;
       free(ptr);
```

void display()

3

4

5

7

8

9

1 2

3

4

9

2

3

5

```
int main()
   int value, choice;
   while (1)
       printf("-----\n");
       printf("1.create_11\n 2.insert_left\n 3.delete\n 4.display\n 5.exit\n");
       scenf("%d", &choice);
       switch (choice)
       case 1:
           create_11();
           break;
       case 2:
           insert_left();
           break;
       case 3:
           de1();
           break;
       case 4:
           display();
           break;
       case 5:
           exit(0);
           break;
       default:
           printf("wrong input!\n");
           break;
```

```
----MENU---
1.create_ll
 2.insert_left
 3.delete
 4.display
 5.exit
1
Enter -1 to exit..
Enter the num: 32
Enter the num: 46
Enter the num: 65
Enter the num: -1
               -MENU-
1.create_ll
 2.insert_left
 3.delete
 4.display
 5.exit
П
32->46->65->-
1.create_ll
2.insert_left
3.delete
4.display
 5.exit
2
enter a value to insert at left:99
Enter the value of node:46
 ------MENU-----
1.create_ll
2.insert_left
3.delete
 4. display
 5.exit
4
32->99->46->65->-----MENU-----
1.create_ll
2.insert_left
3.delete
4.display
```

```
1.create_ll
2.insert_left
3.delete
 4.display
 5.exit
3
enter the value to be deleted:46
1.create_ll
2.insert_left
3.delete
4.display
5.exit
32->99->65->--
1.create_ll
2.insert_left
3.delete
4.display
5.exit
```