```
#include <stdio.h>
#include <stdlib.h>

struct Elm {
	int x;
	struct Elm *next;
	struct Elm *prev;
};
typedef struct Elm Elm;

// List
struct List {
	Elm *head;
	Elm *tail;
	int len;
};
typedef struct List List;

void l_push_back(List *, int);
void l_push_front(List *, int);
void l_insert(List *, int, int);
void l_pop_front(List *);
void l_pop_back(List *);
void l_erase(List *, int);
void l_print(List *);
Elm *l_search(List *, int);

   /* p-ийн зааж буй List-д x утгыг төгсгөлд хийнэ */
   void l_push_back(List *p, int x)
   {
      /* Энд оруулах үйлдлийг хийнэ үү */
      Elm* tmp = malloc(sizeof(Elm));
      tmp->x = x;
      tmp->next = NULL;
      tmp->prev = NULL;
      if(p->len == 0) {
         p->head = p->tail = tmp;
      } else {
         p->tail->next = tmp;
         tmp->prev = p->tail;
         p->tail = tmp;
      }
      p->len++;
   }

   /* p-ийн зааж буй List-д x утгыг эхэнд хийнэ
     Бүх элементүүд нэг нэг байрлал хойшилно.
    */
```

```
void l_push_front(List *p, int x)
{
   /* Энд оруулах үйлдлийг хийнэ үү */
   Elm* tmp = (Elm*)malloc(sizeof(Elm));
   tmp->x = x;
   tmp->next = NULL;
   tmp->prev = NULL;
   if(p->len == 0) {
      p->head = p->tail = tmp;
   } else {
      tmp->next = p->head;
      p->head->prev = tmp;
      p->head = tmp;
   }
   p->len++;
}

/*
  p-ийн зааж буй List-д x утгыг pos байрлалд хийнэ
  pos болон түүнээс хойшхи элементүүд нэг байрлал ухарна.
  Тухайн байрлал List-ийн сүүлийн индексээс их бол төгсгөлд орно.
 */
void l_insert(List *p, int x, int pos)
{
   /* Энд оруулах үйлдлийг хийнэ үү */
   Elm* tmp = (Elm*)malloc(sizeof(Elm));
   tmp->x = x;
   tmp->next = NULL;
   tmp->prev = NULL;

   Elm* hayg;
   if(pos <= 0) {
      tmp->next = p->head;
      p->head = tmp;
   } else if(pos >= p->len) {
      p->tail->next = tmp;
      tmp->prev = p->tail;
      p->tail = tmp;
   } else {
      if(pos <= p->len/2) {
         hayg = p->head;
         for(int i = 0; i < pos - 1; i++) {
            hayg = hayg->next;
         }
         tmp->next = hayg->next;
         hayg->next->prev = tmp;
         tmp->prev = hayg;
         hayg->next = tmp;
```

```
      } else {
         hayg = p->tail;
         for(int i = p->len-1; i > pos; i--) {
            hayg = hayg->prev;
         }
         tmp->next = hayg;
         tmp->prev = hayg->prev;
         hayg->prev->next = tmp;
         hayg->prev = tmp;
      }
   }
   p->len++;
}


/*
  p-ийн зааж буй List-н эхлэлээс гаргана.
  List-ийн бүх элементүүд нэг нэг байрлал урагшилна
 */
void l_pop_front(List *p)
{
   /* Энд гаргах үйлдлийг хийнэ үү */
   if(p->len > 0) {
      Elm* tmp = p->head;
      p->head->next->prev = NULL;
      p->head = p->head->next;
      free(tmp);
      p->len--;
   }
}

/* p-ийн зааж буй List-н төгсгөлөөс гаргана */
void l_pop_back(List *p)
{
   /* Энд гаргах үйлдлийг хийнэ үү */
   if(p->len > 0) {
      Elm* tmp = p->tail;
      p->tail->prev->next = NULL;
      p->tail = p->tail->prev;
      free(tmp);
      p->len--;
   }
}

/* p-ийн зааж буй List-н pos байрлалаас гаргана.
   pos болон түүнээс хойшхи элементүүд нэг байрлал урагшилна.
   pos байрлалаас гарах боломжгүй бол юу ч хийхгүй.
 */
```

```
void l_erase(List *p, int pos)
{
   /* Энд гаргах үйлдлийг хийнэ үү */
   if(pos <= 0) {
      l_pop_front(p);
   } else if(pos >= p->len-1) {
      l_pop_back(p);
   } else {
      Elm* hayg;
      if(pos <= p->len/2) {
         hayg = p->head;
         for(int i = 0; i < pos-1; i++) {
            hayg = hayg->next;
         }
         hayg->prev->next = hayg->next;
         hayg->next->prev = hayg->prev;
         free(hayg);
      } else {
         hayg = p->tail;
         for(int i = p->len-1; i > pos; i--) {
            hayg = hayg->prev;
         }
         hayg->prev->next = hayg->next;
         hayg->next->prev = hayg->prev;
         free(hayg);
      }
      p->len--;
   }
}

/*
  p-ийн зааж буй List-н утгуудыг хэвлэнэ.
  Хамгийн эхний элементээс эхлэн дарааллаар, нэг мөрөнд
  нэг л элемент хэвлэнэ.
 */
void l_print(List *p)
{
      /* Энд хэвлэх үйлдлийг хийнэ үү */
   Elm* tmp = p->head;
   for(int i = 0; i < p->len; i++) {
      printf("%d\n", tmp->x);
      tmp = tmp->next;
   }
}

/*
  p-ийн зааж буй List-с x тоог хайн олдсон хаягийг буцаана.
  Олдохгүй бол NULL хаяг буцаана.
```

```
 */
Elm *l_search(List *p, int x)
{
    Elm* tmp = p->head;
    for(int i = 0; i < p->len; i++) {
        if(tmp->x == x)
            return tmp;
        tmp = tmp->next;
    }
    return NULL;
}

int main()
{
    int x, t, ds, pos;
    List l;
    l.head = l.tail = NULL;
    l.len = 0;

    while (1) {
        printf("1. Stack, 2. Queue, 3. List, 4. Exit\n");
        scanf("%d", &ds);
        if (ds == 4)
            break;
        int done = 0;
        while (!done) {
            printf("1: push (back), 2: pop (back), 3: print, "
                "4: push_front, 5: insert, 6: pop_front, "
                "7: erase, 8: search, 9: exit\n");
            scanf("%d", &t);

            switch (t) {
            case 1:
                printf("Oruulax utga: ");
                scanf("%d", &x);
                if (ds == 1) {}
                    //s_push(&s, x);
                else if (ds == 2) {}
                    //q_push(&q, x);
                else
                    l_push_back(&l, x);
                break;
            case 2:
                if (ds == 1) {}
                    //s_pop(&s);
                else if (ds == 2) {}
                    //q_pop(&q);
                else
```

```
            l_pop_back(&l);
        break;
    case 3:
        if (ds == 1) {}
            //s_print(&s);
        else if (ds == 2) {}
            //q_print(&q);
        else
            l_print(&l);
        break;
    case 4:
        if (ds == 1 || ds == 2)
            printf("Aldaa: Iim uildel xiix bolomjgui\n");
        else {
            printf("Oruulax utga: ");
            scanf("%d", &x);
            l_push_front(&l, x);
        }
        break;
    case 5:
        if (ds == 1 || ds == 2)
            printf("Aldaa: Iim uildel xiix bolomjgui\n");
        else {
            printf("Oruulax utga: ");
            scanf("%d", &x);
            printf("Oruulax bairlal: ");
            scanf("%d", &pos);
            l_insert(&l, x, pos);
        }
        break;
    case 6:
        if (ds == 1)
            printf("Aldaa: Iim uildel xiix bolomjgui\n");
        else {
            if (ds == 2) {}
                //q_pop(&q);
            else
                l_pop_front(&l);
        }
        break;
    case 7:
        if (ds == 1 || ds == 2)
            printf("Aldaa: Iim uildel xiix bolomjgui\n");
        else {
            printf("Gargax bairlal: ");
            scanf("%d", &pos);
            l_erase(&l, pos);
        }
```

```
                break;
            case 8:
                if (ds == 1 || ds == 2)
                    printf("Aldaa: Iim uildel xiix bolomjgui\n");
                else {
                    printf("Xaix utga: ");
                    scanf("%d", &x);
                    Elm *p = l_search(&l, x);
                    if (p == NULL)
                        printf("Oldsongui\n");
                    else
                        printf("Oldson bairlal: %d\n", l_search(&l, x));
                }
                break;
            case 9:
                done = 1;
                break;
            default:
                break;
            }
        }
    }
    return 0;
}
```