

```

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

struct node
{
    struct node *prev;
    int info;
    struct node *next;
};

struct node *addtoempty(struct node *start, int data)
{
    start = NULL;
    struct node *tmp;
    if ((tmp = (struct node *)malloc(sizeof(struct node))) == NULL)
    {
        printf("no mem\n");
        exit(1);
    }
    tmp->info = data;
    tmp->next = NULL;
    tmp->prev = NULL;
    start = tmp;
    return start;
}

struct node *addatbeg(struct node *start, int data)
{
    struct node *tmp;
    if ((tmp = (struct node *)malloc(sizeof(struct node))) == NULL)
    {
        printf("No memory\n");
        exit(1);
    }

    tmp->info = data;
    tmp->next = start;
    tmp->prev = NULL;
    start->prev = tmp;
    start = tmp;
    return start;
};

struct node *addatend(struct node *start, int data)
{
    struct node *tmp, *ptr = start;
    if ((tmp = (struct node *)malloc(sizeof(struct node))) == NULL)
    {
        printf("No mem\n");
        exit(1);
    }
    tmp->info = data;
    while (ptr->next != NULL)
        ptr = ptr->next;
    tmp->next = ptr->next;
    tmp->prev = ptr;
    ptr->next = tmp;
    return start;
}

```

```

struct node *create(struct node *start)
{
    printf("Enter the number of items to be entered: ");
    int n;
    scanf("%d", &n);
    if (n == 0)
        return NULL;
    int data;
    printf("Enter the data to be inserted: ");
    scanf("%d", &data);
    start = addtoempty(start, data);
    for (int i = 2; i <= n; i++)
    {
        printf("Enter the data to be inserted: ");
        scanf("%d", &data);
        start = addatend(start, data);
    }
    return start;
}

```

```

int numnodes(struct node *start)
{
    int count = 0;
    struct node *ptr = start;
    while (ptr != NULL)
    {
        count++;
        ptr = ptr->next;
    }
    return count;
}

```

```

struct node *swapeven(struct node *start)
{
    struct node *ptr = start;
    struct node *head = start, *tail=start;
    while (tail->next != NULL)
        tail = tail->next;
    while (1)
    {
        if (head->next->info==tail->info)
        {
            int temp;
            temp = head->info;
            head->info = tail->info;
            tail->info = temp;
            return start;
        }
        else
        {
            int temp;
            temp = head->info;
            head->info = tail->info;
            tail->info = temp;
            head = head->next;
            tail = tail->prev;
        }
    }
}

```

```

    return start;
}

struct node *swapodd(struct node *start)
{
    if (start == NULL)
    {
        printf("The list is empty\n");
        exit(1);
    }
    struct node *head = start, *tail = start, *ptr = start;
    while (tail->next != NULL)
        tail = tail->next;
    while (1)
    {
        if (head->info == tail->info)
        {
            int temp = head->info;
            head->info = tail->info;
            tail->info = temp;
            return start;
        }
        int temp = head->info;
        head->info = tail->info;
        tail->info = temp;
        head = head->next;
        tail = tail->prev;
    }
}

struct node *swap(struct node *start)
{
    if (start == NULL)
    {
        printf("The list is empty\n");
        exit(1);
    }
    if (numnodes(start) % 2 == 0)
        start = swapeven(start);
    else
        start = swapodd(start);
    return start;
}

void display(struct node *start)
{
    struct node *ptr = start;
    while (ptr != NULL)
    {
        printf("%3d", ptr->info);
        ptr = ptr->next;
    }
}

int main(int argc, char const *argv[])
{
    struct node *start;
    start = create(start);
    printf("The List is: ");

```

```
    display(start);  
    printf("\n");  
    start = swap(start);  
    printf("The List after swapping is: ");  
    display(start);  
    printf("\n");  
    return 0;  
}
```