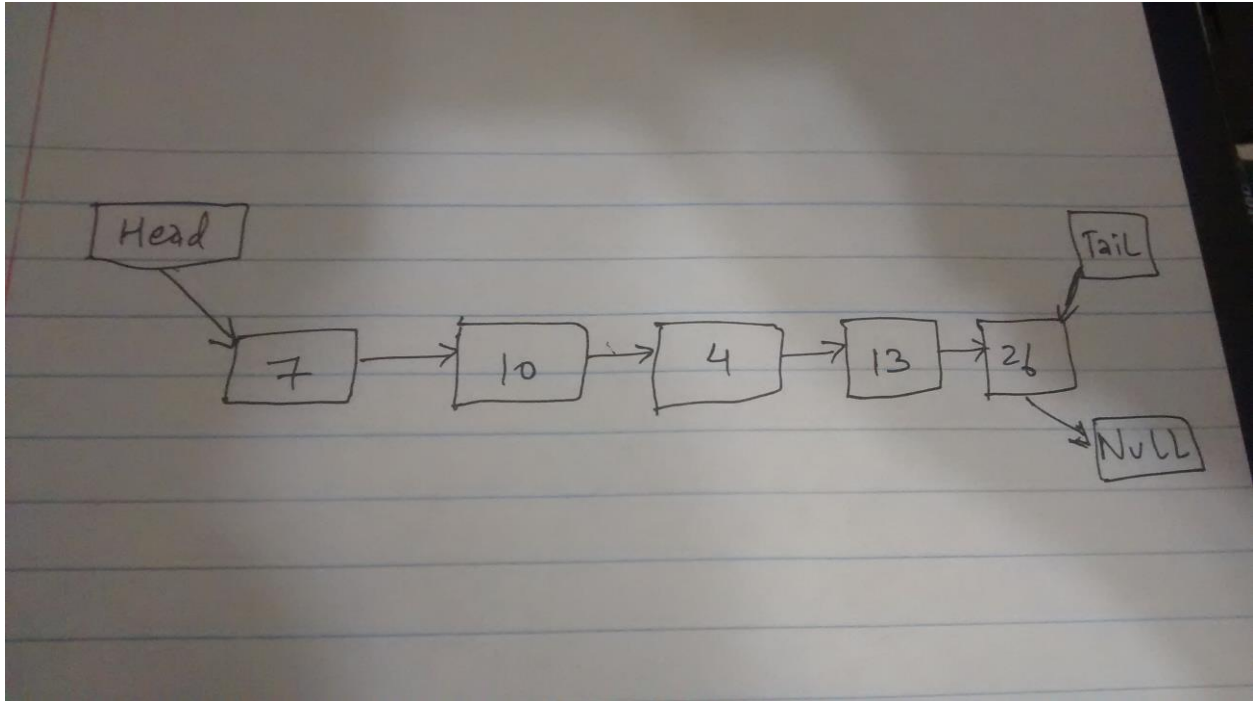


1.



2.

Main function:

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

void addfront(int x);
void pushattheend(int x, struct node *Node4);

int midelement(struct node *head);
void insertatmid(struct node* Node2, struct node* Node3, int x);
struct node{
    int value;
    struct node* next;
};

struct node *head = NULL;

int main(void) {

    struct node *Node1 = (struct node*)malloc(sizeof(struct node));
    struct node *Node2 = (struct node*)malloc(sizeof(struct node));
    struct node *Node3 = (struct node*)malloc(sizeof(struct node));
    struct node *Node4 = (struct node*)malloc(sizeof(struct node));
    struct node *Node5 = (struct node*)malloc(sizeof(struct node));
    struct node *Node0 = (struct node*)malloc(sizeof(struct node));

    head = Node1;
    //head->next = node2;
    Node1->value = 7;
    Node1->next = Node2;
    Node2->value = 10;
    Node2->next = Node3;
    Node3->value = 4;
    Node3->next = Node4;
    Node4->value = 13;
    Node4->next = NULL;

    struct node *comingnode = (struct node*)malloc(sizeof(struct node));
    comingnode = head;

    while (comingnode != NULL)
    {
        printf("%d", comingnode->value);
        comingnode = comingnode->next;
    }

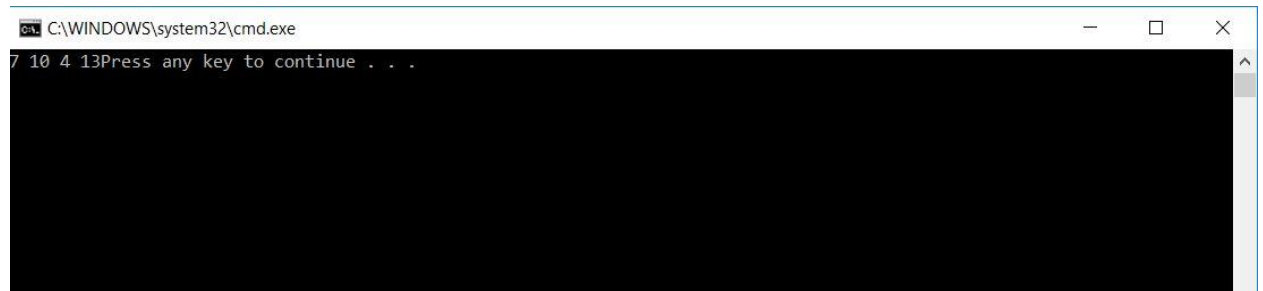
    pushattheend(26, Node4);
    addfront(26);
    int midnumber = midelement(head);
    insertatmid(Node2, Node3, 9);

    return 0;
}
```

Code:

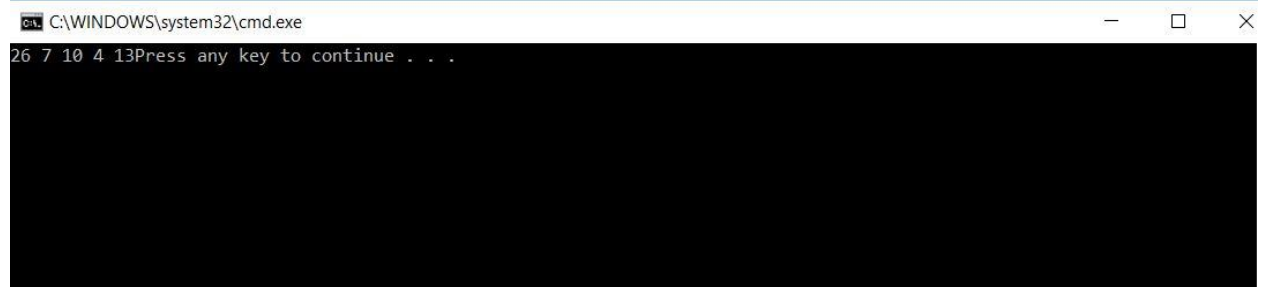
```
void addfront(int x) {  
  
    struct node *Node5 = (struct node*)malloc(sizeof(struct node));  
  
    Node5->value = x;  
    Node5->next = head;  
    head = Node5;  
    return Node5;  
}
```

Before calling the function:



A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The prompt shows the sequence of numbers "7 10 4 13" followed by the text "Press any key to continue . . .". This represents the initial state of a linked list before the addfront function is called.

After calling the function:



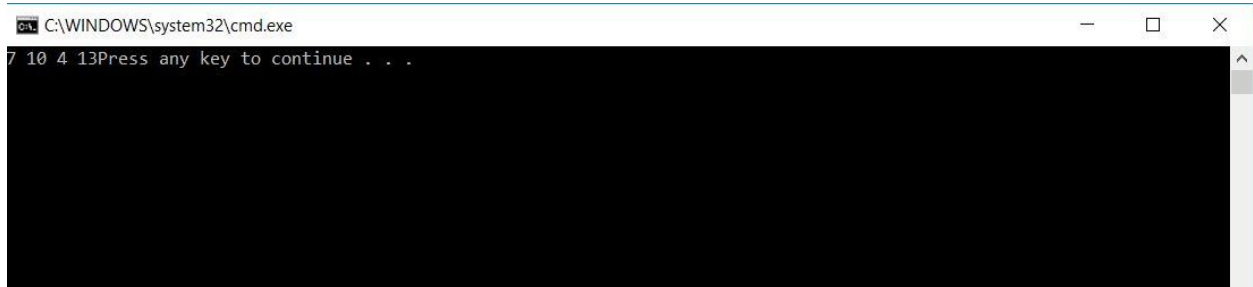
A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The prompt shows the sequence of numbers "26 7 10 4 13" followed by the text "Press any key to continue . . .". This represents the state of the linked list after the addfront function has been called, with the value 26 being added to the front.

3.

Code:

```
void pushattheend(int x, struct node *Node4) {  
    struct node *Node5 = (struct node*)malloc(sizeof(struct node));  
  
    Node5->value = x;  
  
    Node5->next = NULL;  
  
    Node4->next = Node5;  
  
    return;  
}
```

Before calling the function:



```
C:\WINDOWS\system32\cmd.exe  
7 10 4 13Press any key to continue . . .
```

After calling the function:



```
C:\WINDOWS\system32\cmd.exe  
7 10 4 13 26Press any key to continue . . .
```

4.

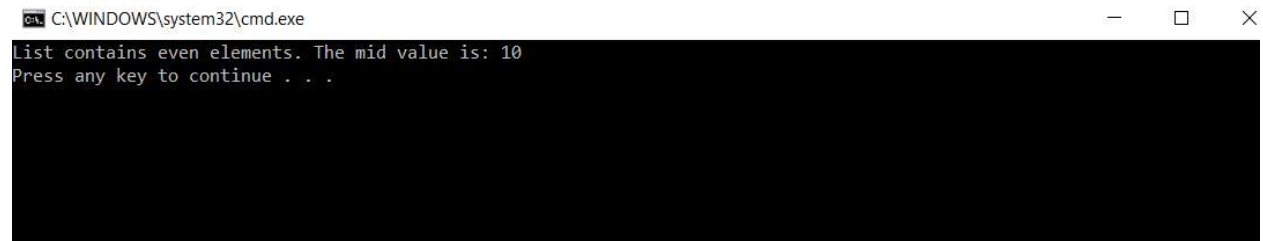
Code:

```
int midelement(struct node *head)
{
    struct node *a, *b;
    int p = 0, count = 0;

    b = a = head;
    //, p->num
    while (b->next != NULL)
        count++;
    {
        b = b->next;
        if (p) a = a->next;
        p = !p;
    }
    if (p) printf("List contains even elements. The mid value is: %d\n", a->next->value);

    else printf("The mid value is: %d\n", a->value);
    return count;
}
```

After calling the function:



A screenshot of a Windows command prompt window. The title bar shows the path "C:\WINDOWS\system32\cmd.exe". The window contains the following text: "List contains even elements. The mid value is: 10" followed by "Press any key to continue . . .".

5.

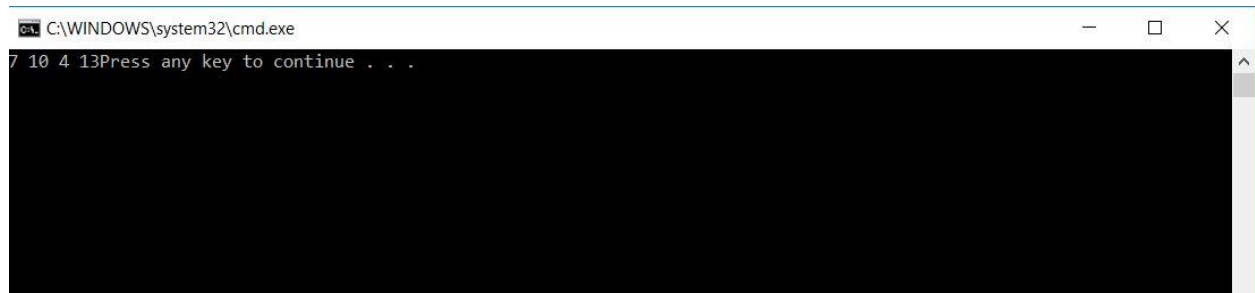
Code:

```
void insertatmid(struct node* Node2, struct node* Node3, int x) {

    struct node *Node6 = (struct node*)malloc(sizeof(struct node));

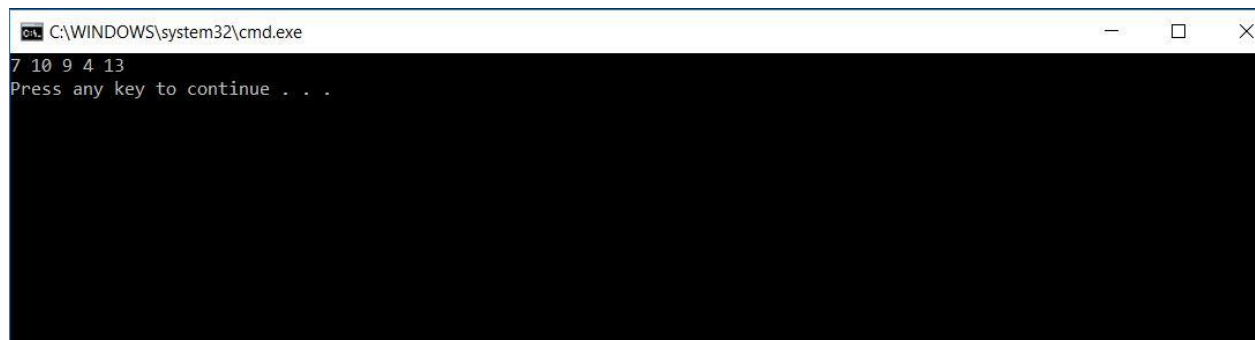
    Node6->value = x;
    Node6->next = Node3;
    Node2->next = Node6;
    return;
}
```

Before calling the function:



```
C:\WINDOWS\system32\cmd.exe
7 10 4 13
Press any key to continue . . .
```

After Calling the function:



```
C:\WINDOWS\system32\cmd.exe
7 10 9 4 13
Press any key to continue . . .
```