Find middle node in linked list

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

struct node
{
	int data;
	struct node *next;
};

void insertAtBeginning(struct node **head, int data)
{
	// allocate new node
	struct node *temp = (struct node*) malloc(sizeof(struct node));
	temp->data = data;
	temp->next = (*head);
	*head = temp;
}

void findMiddleNode(struct node *head)
{
	struct node *mid = head, *temp;
	int count = 0;
	for(temp = head; temp ; temp = temp->next)
	{
		if (count & 1)
			mid = mid->next;
		count += 1;
	}
	if (mid)
		printf("The middle node is = %d", mid->data);
}

int main() {
	struct node *head = NULL;
	insertAtBeginning(&head, 10);
	insertAtBeginning(&head, 20);
	insertAtBeginning(&head, 30);
	insertAtBeginning(&head, 40);
	insertAtBeginning(&head, 50);
	insertAtBeginning(&head, 60);
	insertAtBeginning(&head, 70);
	findMiddleNode(head);
	return 1;
}
```

Time complexity: O(n)
Space complexity: O(1)