Node

```
1  struct node
2  {
3     int entry;
4     struct node *next;
5  };
```

Making the head

```
1  struct node * make_head(int new_entry)
2  {
3     struct node * head;
4     head=(struct node *)malloc(sizeof(struct node));
5     head->entry=new_entry;
6     return head;
7  }
```

Print

```
1  void print_list(struct node * iterator)
2  {
3     while(iterator->next!=NULL)
4        {
5           printf("%d\n",iterator->entry);
6           iterator=iterator->next;
7        }
8
9     printf("%d\n",iterator->entry);
10 }
```

Locate an entry

```
1  struct node * locate(struct node * it, int num)
2  {
3     while(it->next!=NULL)
4        {
5           if(it->entry==num)
6              return it;
7           it=it->next;
8        }
9     if(it->entry==num)
10       return it;
11
12    return NULL;
13 }
```

Adding a node

```
1  void add_node(struct node * here, int new_entry)
2  {
3     struct node * here_next=here->next;
4     here->next = (struct node *)malloc(sizeof(struct
5     here->next->next=here_next;
6     here->next->entry=new_entry;
7  }
```

Appending a node

```
1  void append_node(struct node * head, int new_entry)
2  {
3     while(head->next!=NULL)
4        head=head->next;
5     head->next =
6        (struct node *)malloc(sizeof(struct node));
7     head->next->entry=new_entry;
8  }
```

Deleting the next node

```
1  void delete_next(struct node * here)
2  {
3     if(here->next==NULL)
4        return;
5
6     struct node * temp=here->next;
7     here->next=here->next->next;
8     free(temp);
9  }
```