Linked list sheet

1-

```
Void list_head_insert(node*& head_ptr, const node::value_type& entry)
{
Node *insert_ptr;
Insert_ptr = new node;
Insert_ptr->set_data(entry);
Insert_ptr->set_link(head_ptr);
Head_ptr = insert_ptr;
}
```

2-

Create a temporary variable of type class node and assigns p->next to the temporary variable – call this variable temp , Assign temp's next pointer to p's next pointer , Set the next pointer of the node after *p to NULL.

3-

```
Typedef struct Node;
Typedef Node* NodePtr;
Struct Node{
   Int x;
   NodePtr next;
};
Int main ()
```

```
{
   Int n;
   NodePtr head, ptr = NULL;
   Head = ptr;
   While (cin >> n){
      Ptr = new Node;
      Ptr->x = n;
      Ptr->next = NULL;
      Ptr = ptr->next;
   }
   NodePtr bling = head;
   While(bling != NULL){
      Cout << bling->x << endl;
      Bling = bling->next;
   }
   Return 0;
}
```

4-

```
Insert_ptr->set_link (previous_ptr->link() );
Previous_ptr->set_link (insert_ptr);
```

6-

```
Int sum_of_nodes(Node *head_ptr)
{
Node *p;
```

```
Int sum = 0;


For (p = head_ptr; p!= NULL; p = p->link)
Sum = sum + p->data;
Return sum;
}
```

7-

```
Int product_of_nodes(Node *head_ptr)
{
Node *p;
Int product = 1;
For (p = head_ptr; p!= NULL; p = p->link)
Product = product * p->data;
Return product;
}
```