Tutorial Sheet - 7

Date...../...../.....

Page 7.1

1.
```
for (i=0; i<MAX_SIZE; i++)
{
    temp1 = temp1 → next;
    for (j=0; j<MAXsize; j++)
    {
        temp2 = temp2 → next;
        if (temp1 → next == temp2 → next)
        {
            printf ("%ud is point of int", *temp1 → next)
        }
    }
}
```

3.
```
// function to be called
node * reverse (node* header)
{
    node *prev = NULL;
    node * temp = head;
    node * next = 0
    while (current != NULL)
    {
        next = current → next;
        current → next = prev;
        prev = current;
        current = next;
    }
    return * prev;
}
```

4. ~~temp = temp → next;~~
~~count ++;~~
for (i=0; i< MAX_SIZE; i++)
{
~~temp = temp → next;~~
~~count++;~~
if (count % 2 == 0)
{
temp → next = prev;
& prev = temp;
~~temp = temp → next;~~
}
temp = temp → next;
count ++;
}

// let prev be defined as head at start.

6. for (i=0; i< MAX; i++)
{
if (temp → next != NULL)
{
temp = temp → next;
count ++;
}
}
temp = head;
for (i = count; i >= 0; i--) NEXT PAGE
{
if (count > m)
{
break;
count ++;
}
}

~~else if (count == 0)~~
~~{~~
~~temp → next = head;~~

```
temp = head;
for (i=count; i>=0; i--)
{
  if (i > count - m)
  {
    temp = temp → next;
  }
  else if (i == 0)
  {
    temp → next = head;
  }
  else if (i == count-m)
  {
    temp → next = NULL;
  }
  else
  {
    temp = temp → next;
  }
}
```

7. h1.
   = reverse (head); // calling function from Q3.

```
temp = h1;                    carry = 1;
for(i=0; i<size; i++)
{
  if ( temp → data == 9)
```

```
        temp->data = 0;
        carry = 1;
    }
    else
    {
        temp->data = temp->data + carry;
        carry = 0;
    }
}
```

2.
```
if (x == y)
{
    exit(0);
}
else
{
    for (i = 0; i < MAX_SIZE; i++)
    {
        if (temp->data == x || temp->data == y)
        {
            nodex = temp;
            x1 = i;
        }
        else if (temp->data == y)
        {
            y1 = i;
            nodey = temp;
        }
        temp = head;
    }
    for (i = 0; i < MAX; i++)
    {
        if (temp == nodex || temp ==
```

```
    }
        temp->next = nodey->next;
    }
else if (temp == nodey)
    {
        temp->next = nodex->next;
    }
else if ( temp->next ==nodey)
    {
        temp->next = nodex; }
else if ( temp->next == nodex)
    {
        temp->next == nodey;
    }
```