```
    ptr1 = ptr;
    ptr = ptr -> link;
  }
  ptr1 -> link = NULL;
  }
  printf("\n Deleted element is : %d", *ptr);
}
```

Assignment - 4

Q In the binary search algorithm, it is suggested to calculate the mid as beg + (end - beg)/2 instead of (beg + end)/2. Why is it so?

Beg (beg + end) may overflow which then means you get a result that is less than beg or far into the negative if you are using signed integer. So, instead they take the distance b/w beg & end & add half of that to beg. This is only a single extra operation to make the algorithm more robust.

Q) Write the algorithm/ function for ternary search

```
int ternary_search(int l, int r, int x)
{
  if(r >= l)
  {
    int mid1 = l + (r - l)/3;
```

```
int mid2 = r - (r-l)/3;
    if (ar[mid1] == x)
        return mid1;
    if (ar[mid2] == x)
        return mid2;
    if (x < ar[mid1])
return ternary_search(l, mid1-1, x);
else if (x > ar[mid2])
return ternary_search(mid2+1, r, x);
else return ternary_search(mid1+1, mid2-1, x)
}
return -1;
}
```