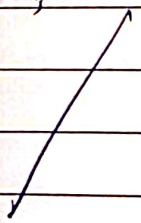
2) Create or simulate a linked list using C with (create & display & delete from start, end, at any position given).

```
void create ()
{
char ch ;
node *new1, *curr;
do {
 new1 = (node *) malloc (sizeof (node));
 printf ("\n enter value :\n");
 if (start == NULL)
  {
   start = new1;
   curr = new1;
  }
 printf ("do you want to Add an element (Y/N) ?");
  scanf ("%c", &ch);
}
while (ch == 'Y' || ch == 'y');
curr -> link = NULL;
}
```



```
Void display () {
  if (start == NULL)
  printf ("\n Linked list is empty \n");
   return;
  }
  node * temp = start;
  printf ("\n element in linkend list : \n");

  while (temp != NULL) {
   printf ("%d ", temp -> data);
   temp = temp -> link;
   }
   printf ("\n");
}
Void Delete from start (){
   if (start == NULL) {
      printf ("Linked list is empty\n");
       return;
      }
   node * temp = start;
    start = start -> link;
    free (temp);
    printf ("\n first element deleted successfully");
}
```

```
Void delete at position (){
    int pos, i = 1;
  if (start = = NULL ) {
  printf ("\n Linked list is empty \n");
   return ;
  }
  printf ("\n Enter the position to delete : ");
  scanf ("%d "&pos );
  node * temp = start;
  node * prev = NULL;
 if (pos == 1 ) {
     start = temp -> link;
     free (temp),
   printf ("\n element at position %d deleted
        Successfully \n ", pos );
   }
while (temp ! = NULL && i < pos) {
   prev = temp;
   temp = temp -> link;
   i++;
  }
if (temp == NULL ){
   printf ("\n position not found \n ");
   return ;
  prev -> link = temp -> link;
   free (temp);
   printf ("\n element at position %d delete
     successfully \n", pos );
```

```
Void DeleteatEnd(){
  if (start == NULL) {
  printf("\n Linked List is empty.\n");
   return;
}
 node *temp = start;
 node * prev = NULL;

 if (start -> link == NULL) {
  printf("Linked List is empty\n");
   return;
}      start = NULL;
 node * free (temp);
     printf("\n Last element deleted successfully \n");
     return;
   }
while (temp -> link != NULL){
    prev = temp;
     temp = temp -> link ;
  }
  prev -> link = NULL;
  free (temp);
  printf("\n Last element deleted successfully \n");
}
```

Output.

1) create
2) display
3) delete from beginning
4) delete at position
5) delete at end
6) exit.

Enter choice: 1
Enter value: 5
Do you want to continue: Y
Enter value: 10
Do you want to continue: Y
Enter value: 15
Do you want to continue: Y
Enter value: 20
Do you want to continue: Y
Enter value: 25
Enter choice: 2
Entered are:
5 10 15 20 25
Enter choice: 3
First element deleted
Enter choice: 2
Elements
10 15 20 25
Enter choice: 4
Enter position to delete: 2
Element at position 2 deleted
Enter choice: 2
Elements are
10 20 25

Enter choice: 5
Last element deleted
Enter choice: 2
10 20
Enter choice: 6