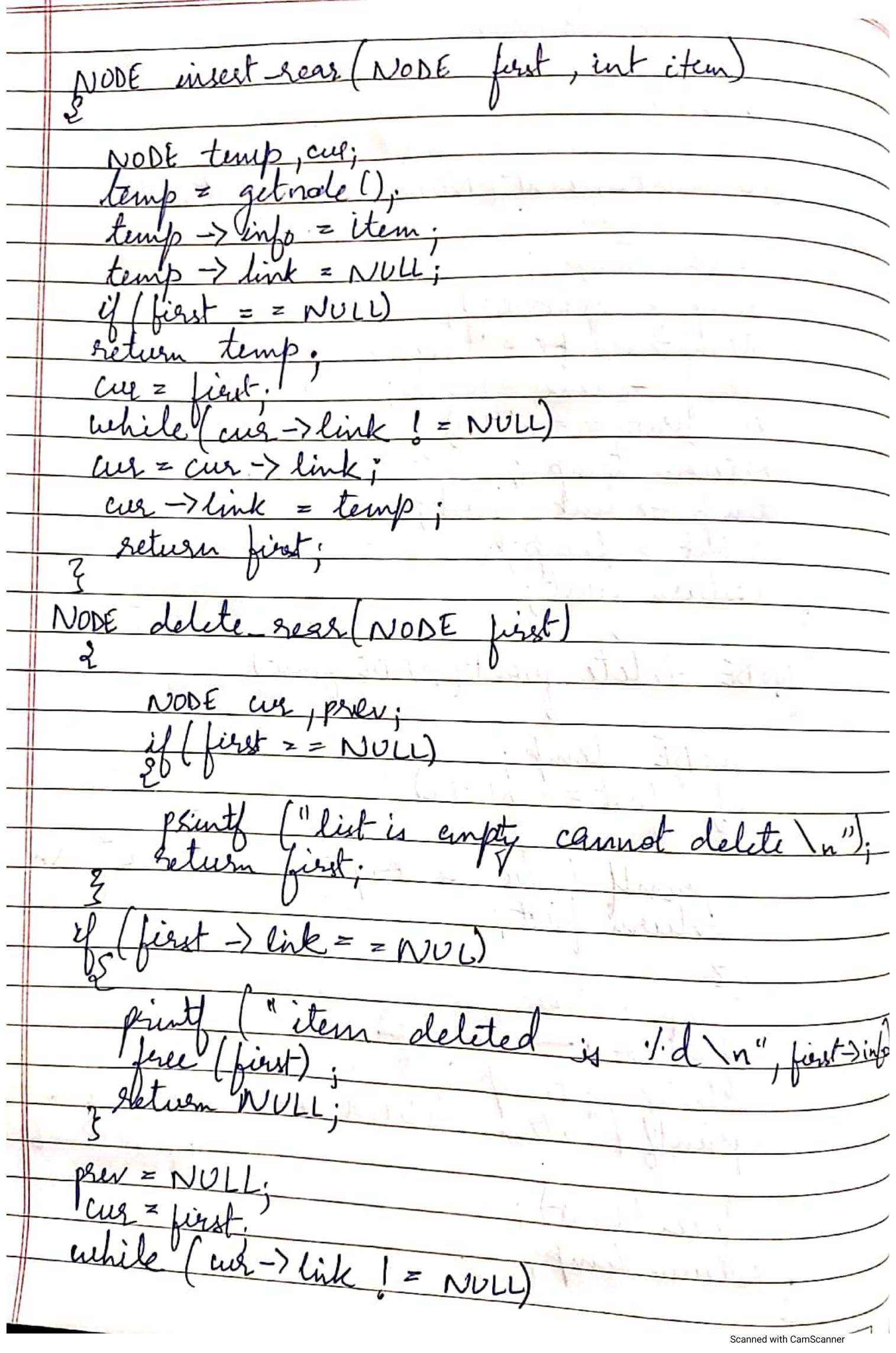
Date ___ /___ /_ to implement simply ellaving operation: uplay the content of the linked # insest (stdio.h) insuf Stallib. h include (conio.h) struct node = WODE) mallor (size Scanned with CamScanner

noid freendle (NODE X) NODE insest-front (NODE first, intitem) Scanned with CamScanner



cus = cus -> link? quit ("item deleted ast rear end is 1.d", eur. Disto). ink = NULL; NODE delete-pos (int. pos, NODE first) NODE peur, cus; first == NULL 11 pos (=0) "Invalid position \n" PBSZZ

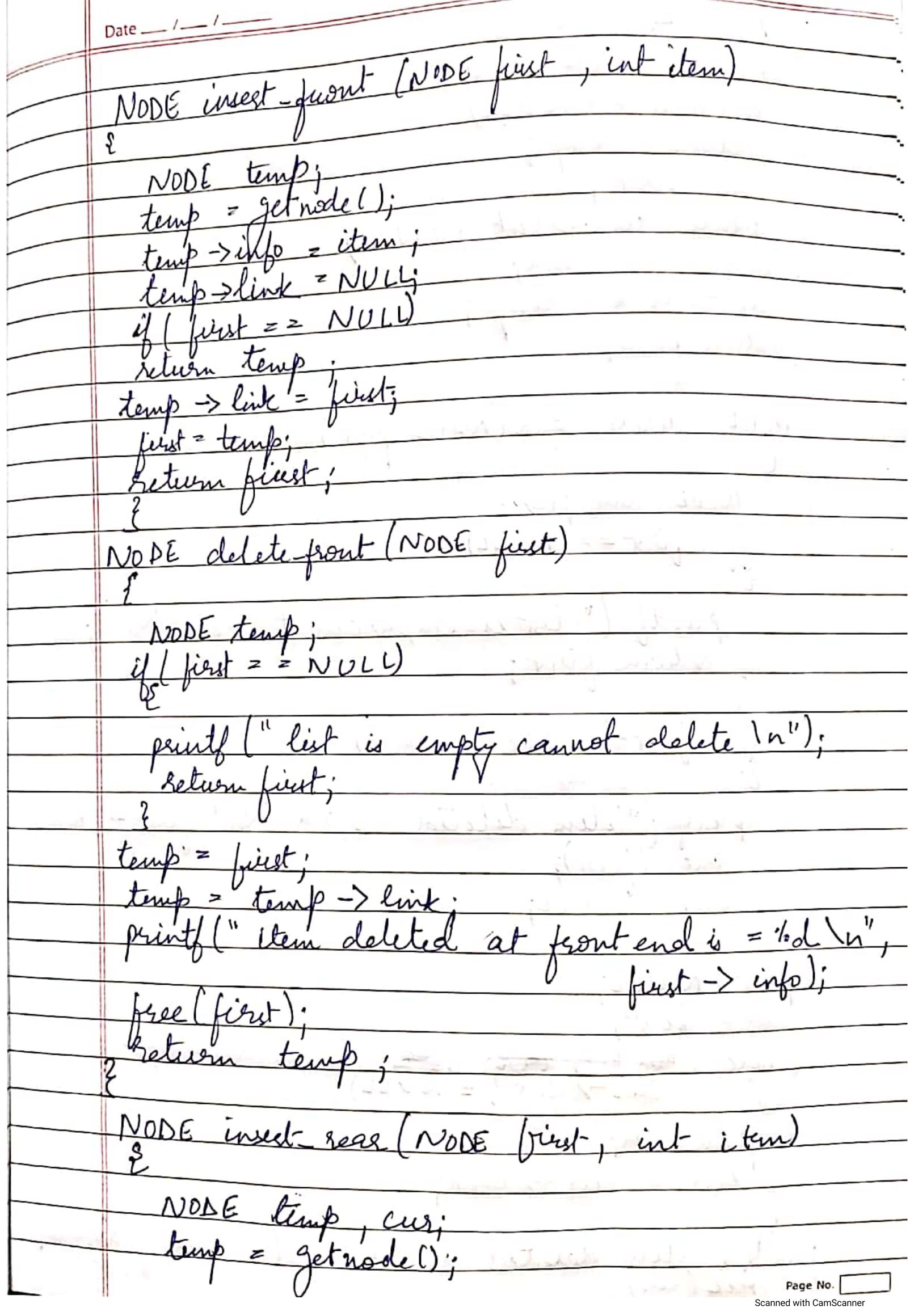
Date ___ / __ / -2 break; Prox = cus; > link; count ++1 if (count = pos) prints ("Invalid position \n"), Peer -> link & cur > link; temb = gétnode l' = 2 NULL & Spos= return temp;

Scanned with CamScanner

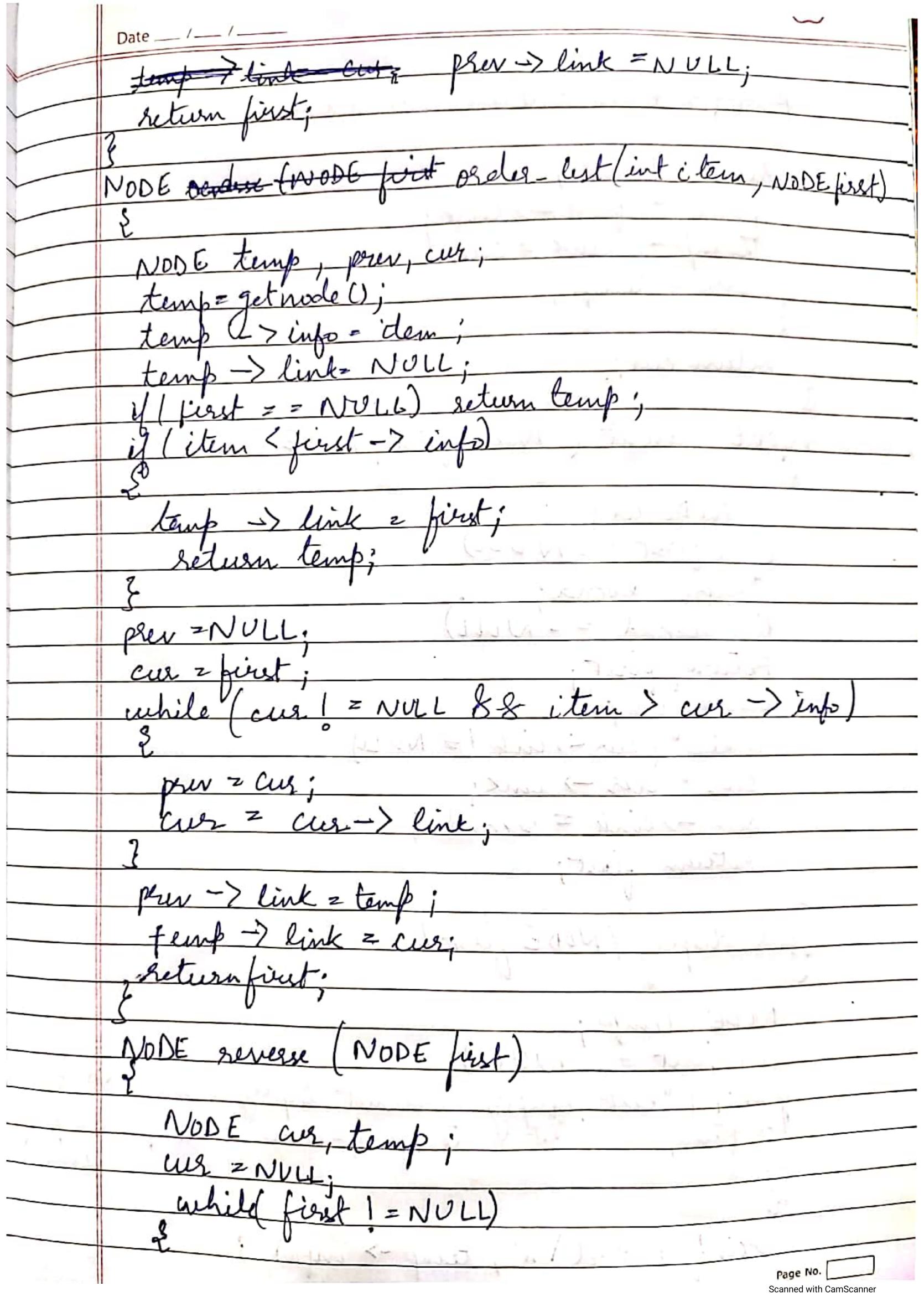
per = = 1) temp -> link = first; setum temp; ar = piet; while (cur! = NULL && count! = pos) cur = cur; count ++; count = = pos) pew -> lik = temp; NODE tenp (= NULL & temp-temp-) (m); Scanned with CamScanner

void main () int item, choice, pos; NODE fierst = NULL; print ("\n 1: Insert-front \n 2: Delite front \n 3: Insert-regar \n 45. Delite at spec rear \n 5. Delete at specified position In 6: Insert at specified position INT: Displan "enter the choice In scanf ("./.d" & choice); Scanned with CamScanner

WAP Implement Single link list with following operation & the linked list Concatenation of two linked include (stdio. h) include (conio. h) Typedel struct node * NODE; NODE getnode & X; X = (NODE) mallor (size of (street node)); if (X = = NULL) erients ("memory full \n"). void freenæle (NODE X) Scanned with CamScanner



temp -> info = item; temp -> link = NULL; seturn temp; cus = first; while (cur-> link ! = NULL) Cur = cur -> link; Cus-> link = temp; return first; NODE delete sear (NODE first) scint ("list is empty cannot delete"
return first; 1: 1



temp -> link; cus = temp; seturn cus; NODE concat (NODE great, NODE second) NODECUS; (first = = NULL) = = NULL) lile (cur-> link 1=. NULL) cus = tus -> link; first Vi temp 1 = NULL; temp= temp=> paint/ ("of.d)n", temp -> info) 3 Scanned with CamScanner

void main (: Insert front In 2: Delete. south (hoice) case 1: prints ("enter the item at front end in)
scanf ["/d", & item);
first = insect front (first, item); first = delete - front (first); grint ("enter the item of (")'d", bitem); insest rear (first, item); delite_ rear i'enter the item to be inverted in ordered List \n"); Scanned with CamScanner

Cove 7: Print ("Enter the no. of Sear ("I'd", Su); Scanned with CamScanner