**LABORATORY PROGRAM – 6**

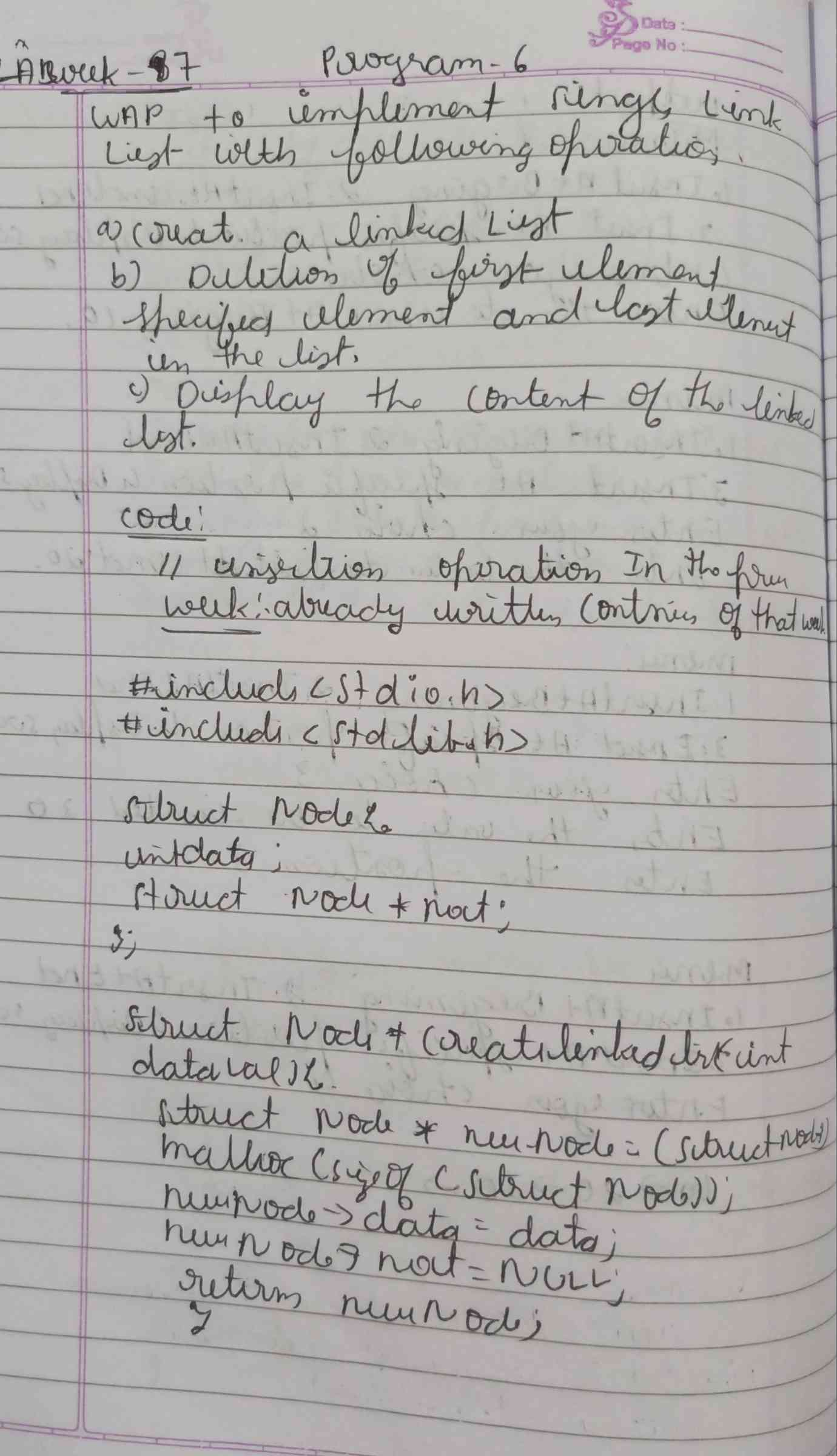
WAP to Implement Singly Linked List with following operations

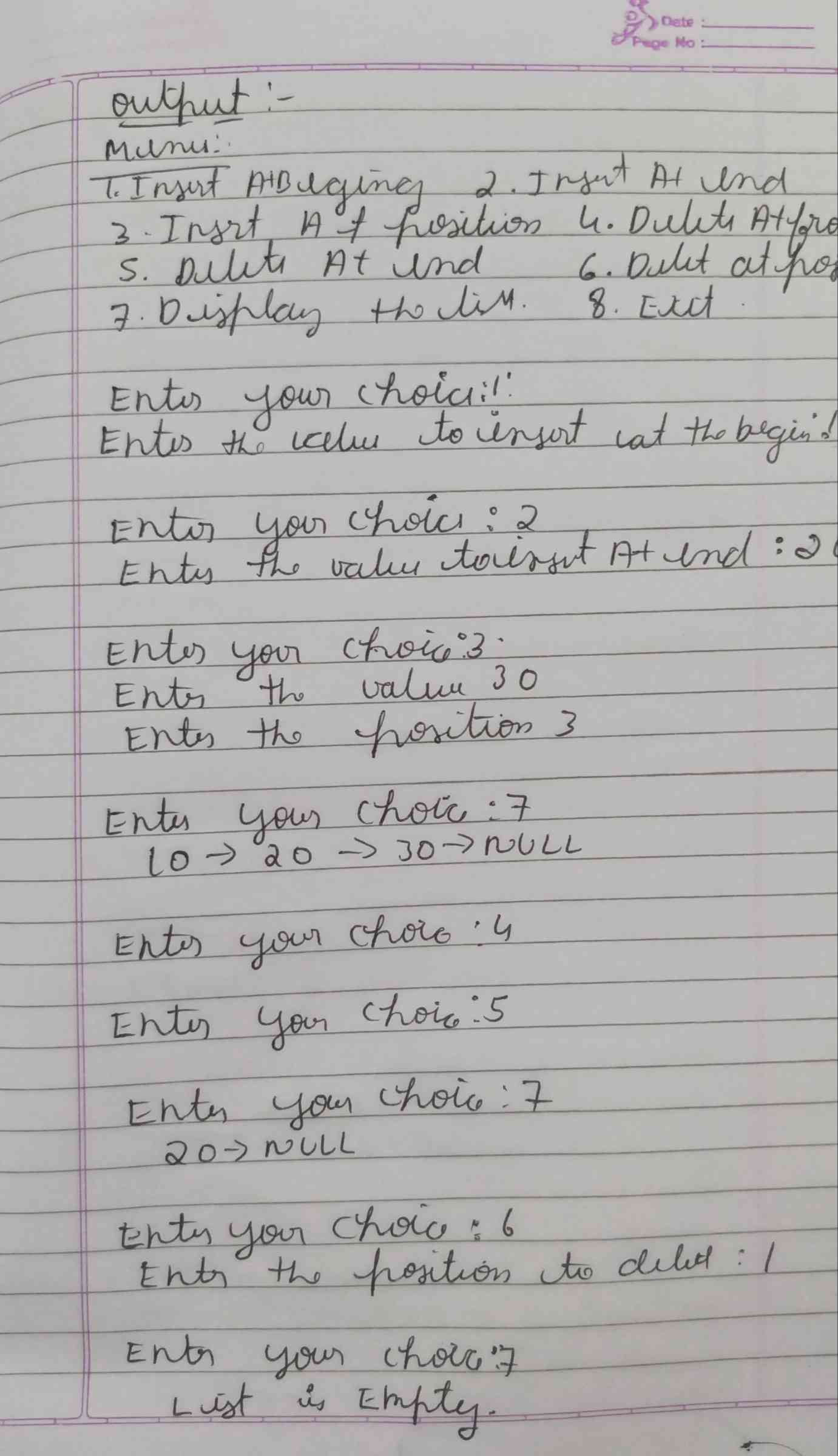
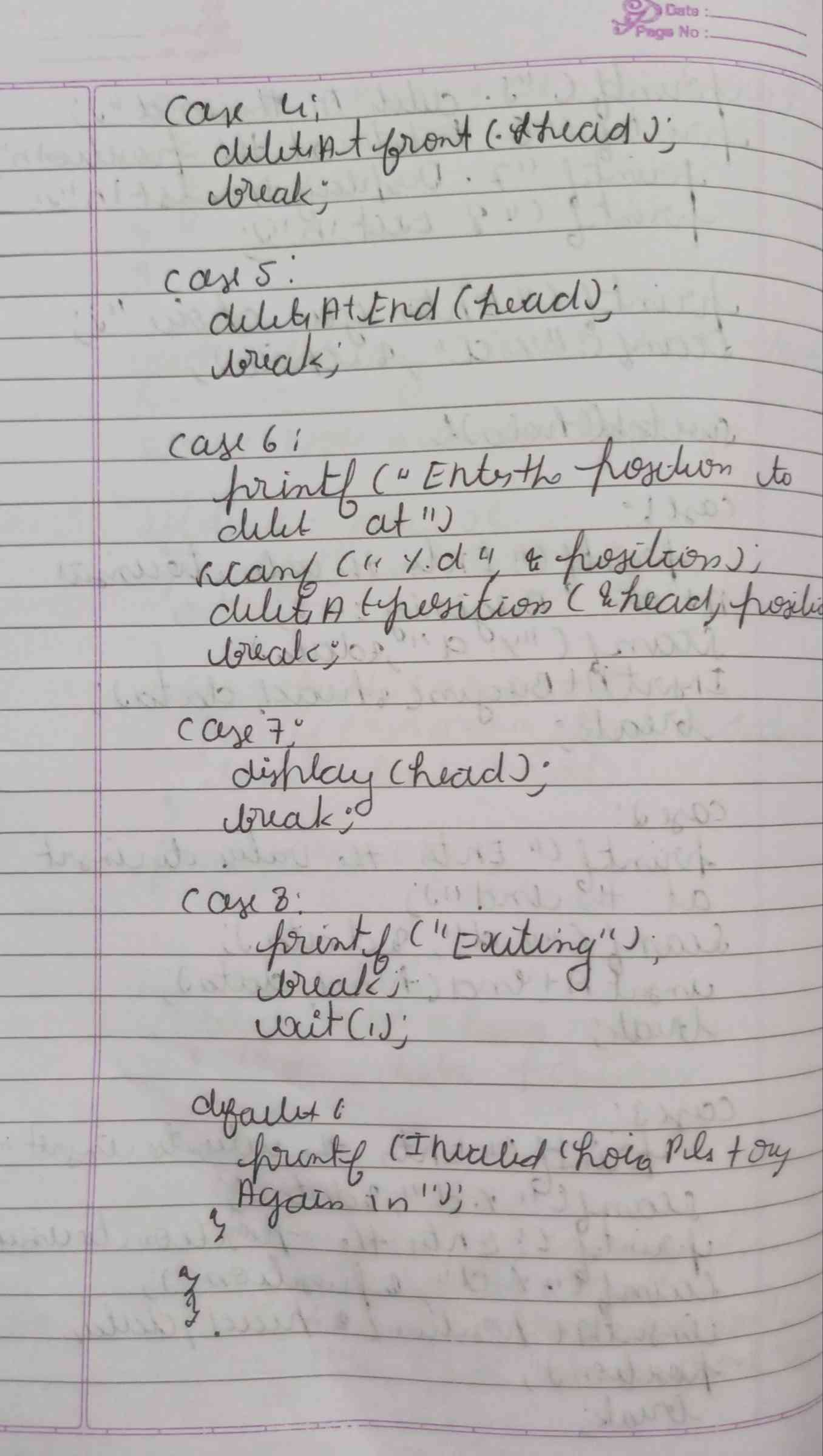
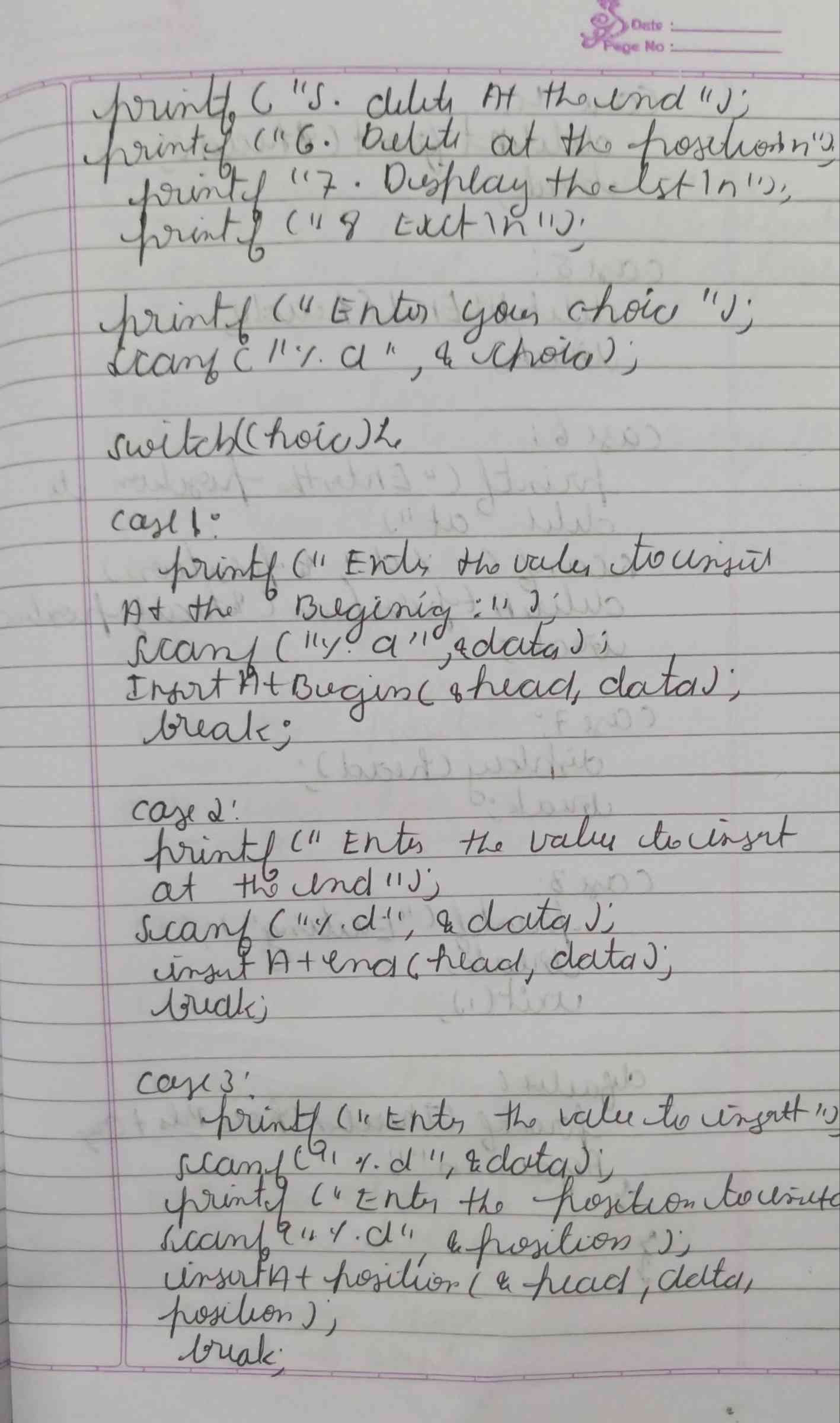
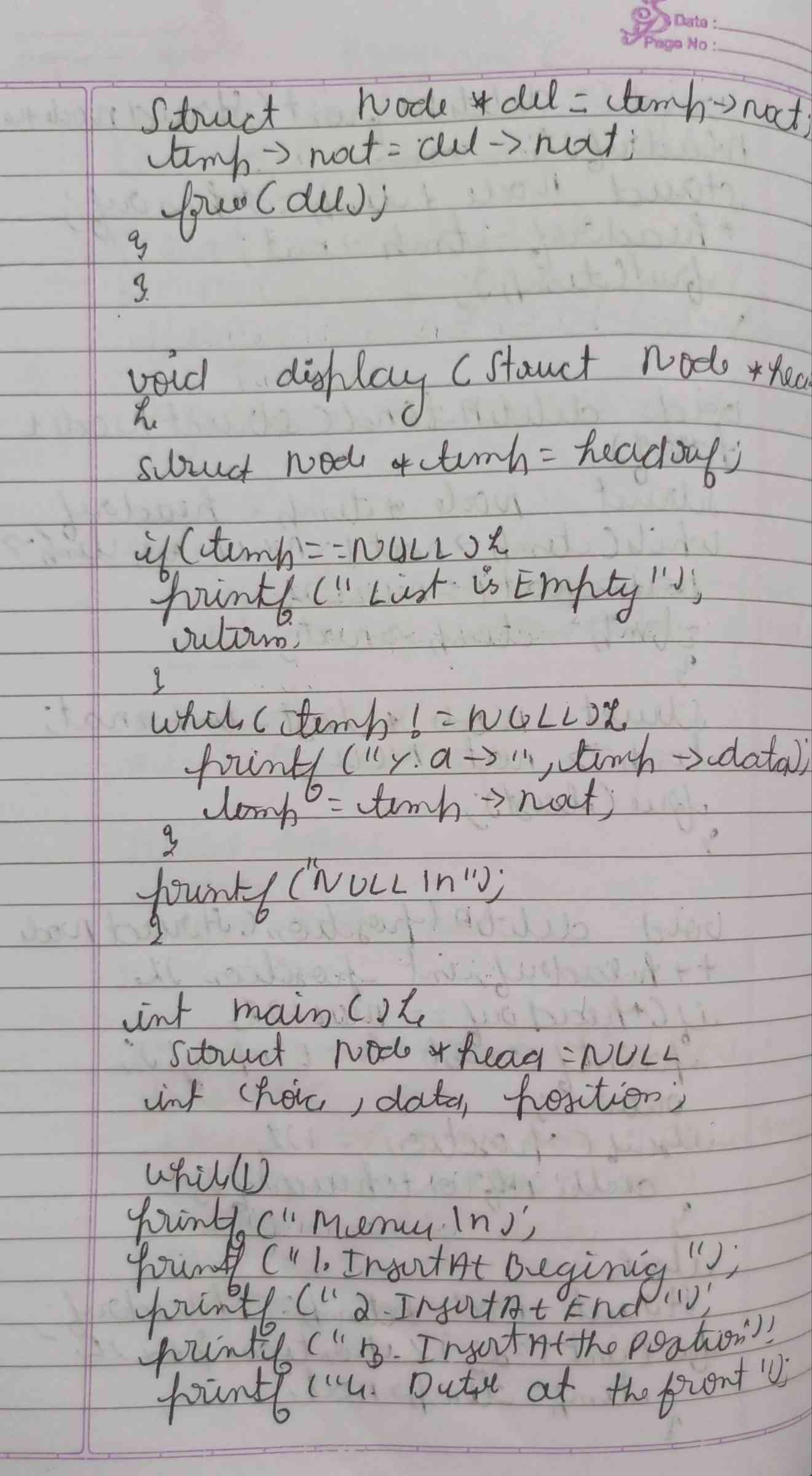
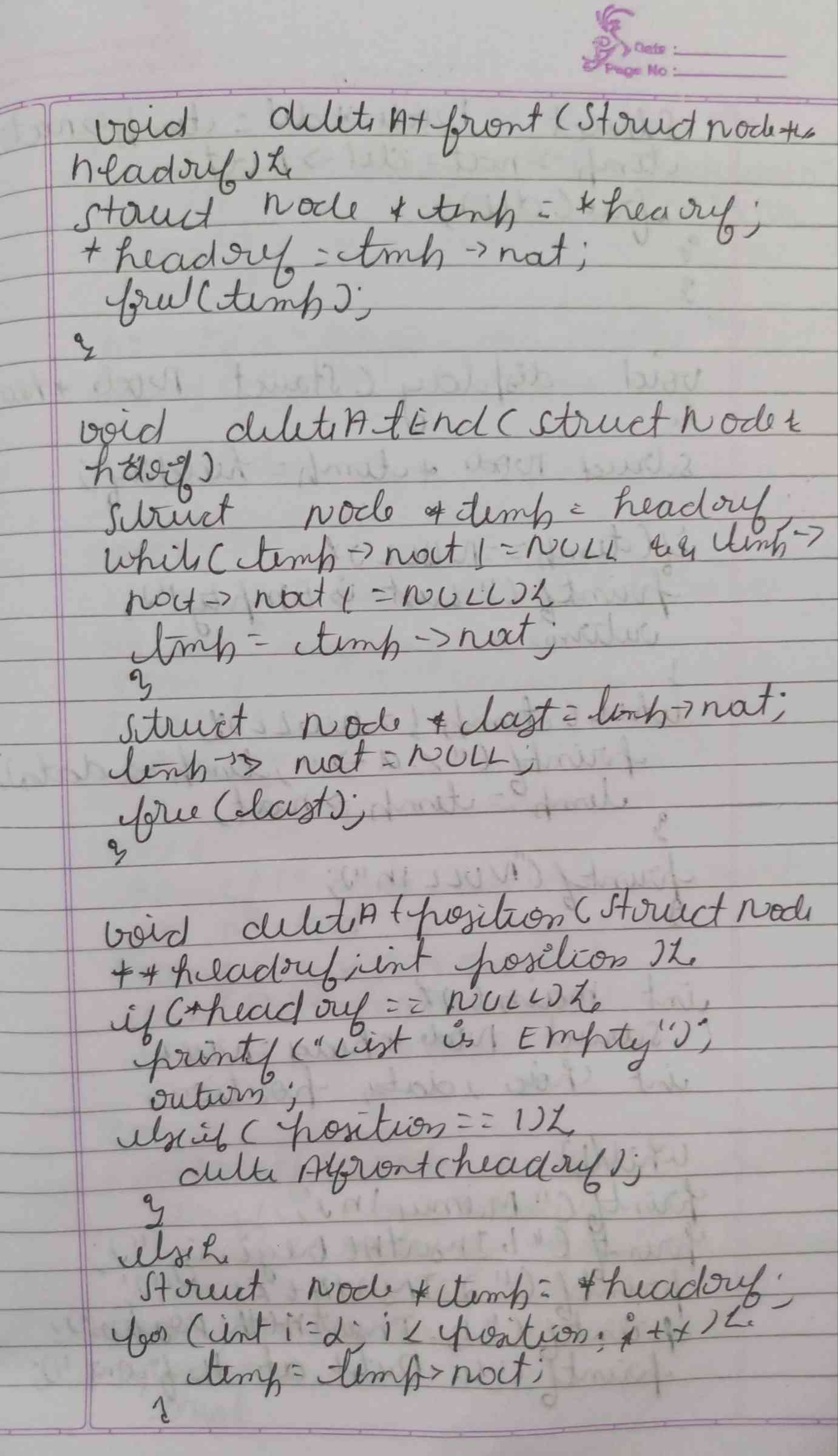
a) Create a linked list.

b) Deletion of first element, specified element and last element in the list.

c) Display the contents of the linked list

**OBSERVATION :**

****

****

**CODE :**

**#include <stdio.h>**

**#include <stdlib.h>**

**struct Node {**

**int data;**

**struct Node\* next;**

**};**

**void createList(struct Node\*\* head) {**

**\*head = NULL;**

**}**

**void deleteFirst(struct Node\*\* head) {**

**if (\*head == NULL) {**

**printf("The list is empty. No element to delete.\n");**

**return;**

**}**

**struct Node\* temp = \*head;**

**\*head = (\*head)->next;**

**free(temp);**

**printf("First element deleted successfully.\n");**

**}**

**void deleteElement(struct Node\*\* head, int value) {**

**if (\*head == NULL) {**

**printf("The list is empty. No element to delete.\n");**

**return;**

**}**

**struct Node\* temp = \*head;**

**struct Node\* prev = NULL;**

**if (temp != NULL && temp->data == value) {**

**\*head = temp->next;**

**free(temp);**

**printf("Element %d deleted successfully.\n", value);**

**return;**

**}**

**while (temp != NULL && temp->data != value) {**

**prev = temp;**

**temp = temp->next;**

**}**

**if (temp == NULL) {**

**printf("Element %d not found in the list.\n", value);**

**return;**

**}**

**prev->next = temp->next;**

**free(temp);**

**printf("Element %d deleted successfully.\n", value);**

**}**

**void deleteLast(struct Node\*\* head) {**

**if (\*head == NULL) {**

**printf("The list is empty. No element to delete.\n");**

**return;**

**}**

**if ((\*head)->next == NULL) {**

**free(\*head);**

**\*head = NULL;**

**printf("Last element deleted successfully.\n");**

**return;**

**}**

**struct Node\* temp = \*head;**

**while (temp->next != NULL && temp->next->next != NULL) {**

**temp = temp->next;**

**}**

**free(temp->next);**

**temp->next = NULL;**

**printf("Last element deleted successfully.\n");**

**}**

**void displayList(struct Node\* head) {**

**if (head == NULL) {**

**printf("The list is empty.\n");**

**return;**

**}**

**struct Node\* temp = head;**

**printf("Linked List: ");**

**while (temp != NULL) {**

**printf("%d -> ", temp->data);**

**temp = temp->next;**

**}**

**printf("NULL\n");**

**}**

**void insertAtEnd(struct Node\*\* head, int value) {**

**struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node));**

**newNode->data = value;**

**newNode->next = NULL;**

**if (\*head == NULL) {**

**\*head = newNode;**

**} else {**

**struct Node\* temp = \*head;**

**while (temp->next != NULL) {**

**temp = temp->next;**

**}**

**temp->next = newNode;**

**}**

**printf("Node with value %d inserted at the end.\n", value);**

**}**

**int main() {**

**struct Node\* head;**

**createList(&head);**

**int choice, value;**

**while (1) {**

**printf("\nSingly Linked List Operations:\n");**

**printf("1. Insert at End\n");**

**printf("2. Delete First Element\n");**

**printf("3. Delete Specified Element\n");**

**printf("4. Delete Last Element\n");**

**printf("5. Display the list\n");**

**printf("6. Exit\n");**

**printf("Enter your choice: ");**

**scanf("%d", &choice);**

**switch (choice) {**

**case 1:**

**printf("Enter the value to insert at the end: ");**

**scanf("%d", &value);**

**insertAtEnd(&head, value);**

**break;**

**case 2:**

**deleteFirst(&head);**

**break;**

**case 3:**

**printf("Enter the value to delete: ");**

**scanf("%d", &value);**

**deleteElement(&head, value);**

**break;**

**case 4:**

**deleteLast(&head);**

**break;**

**case 5:**

**displayList(head);**

**break;**

**case 6:**

**exit(0);**

**default:**

**printf("Invalid choice! Please try again.\n");**

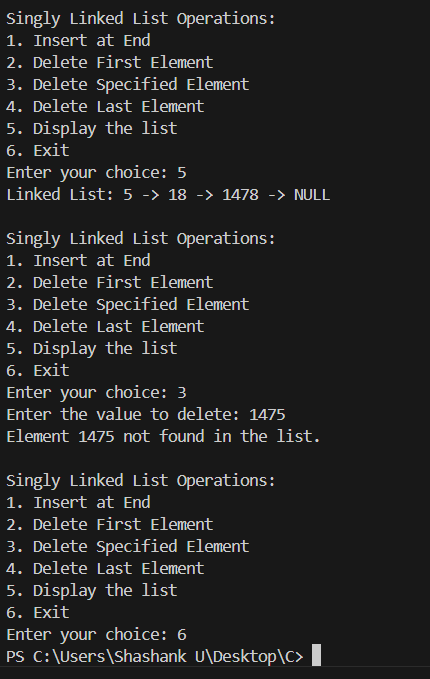
**}**

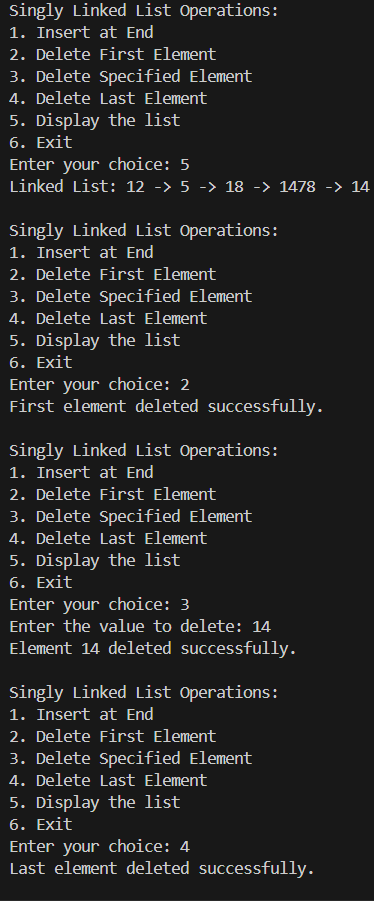
**}**

**return 0;**

**}**

**OUTPUT :**





**Leetcode problem(Daily Temperatures):**

