**LABORATORY PROGRAM – 5**

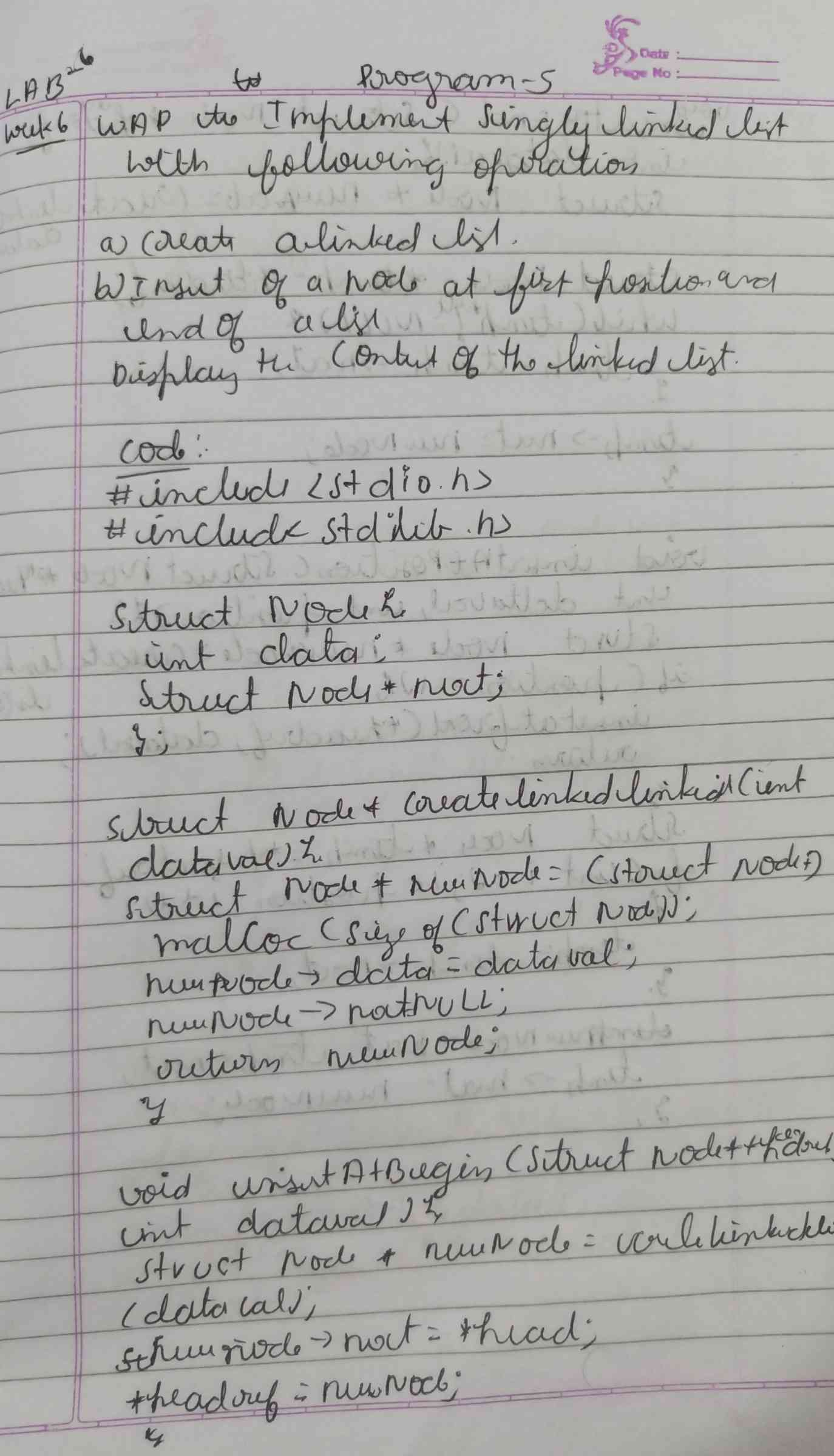
5a)WAP to Implement Singly Linked List with following operations

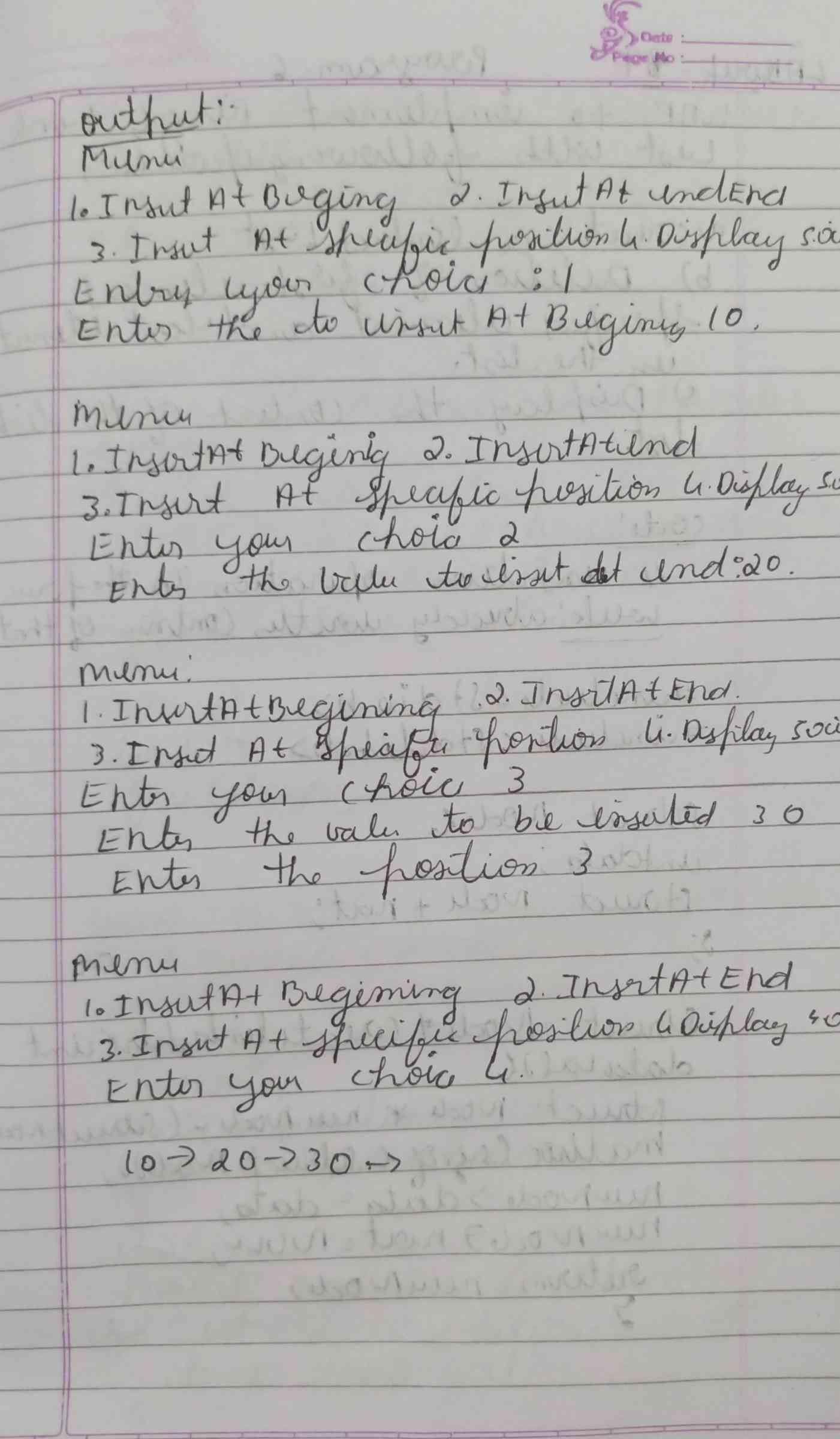
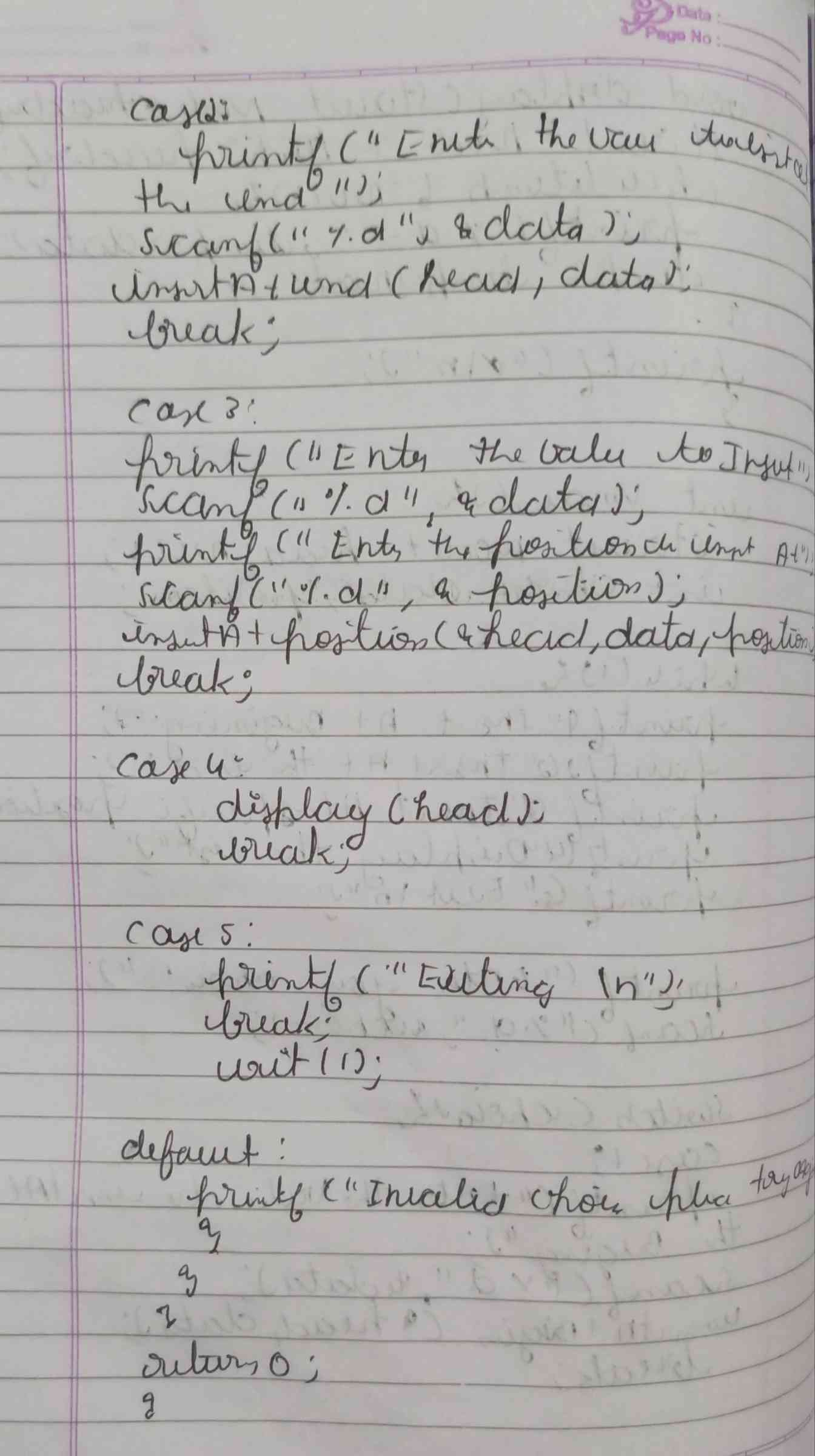
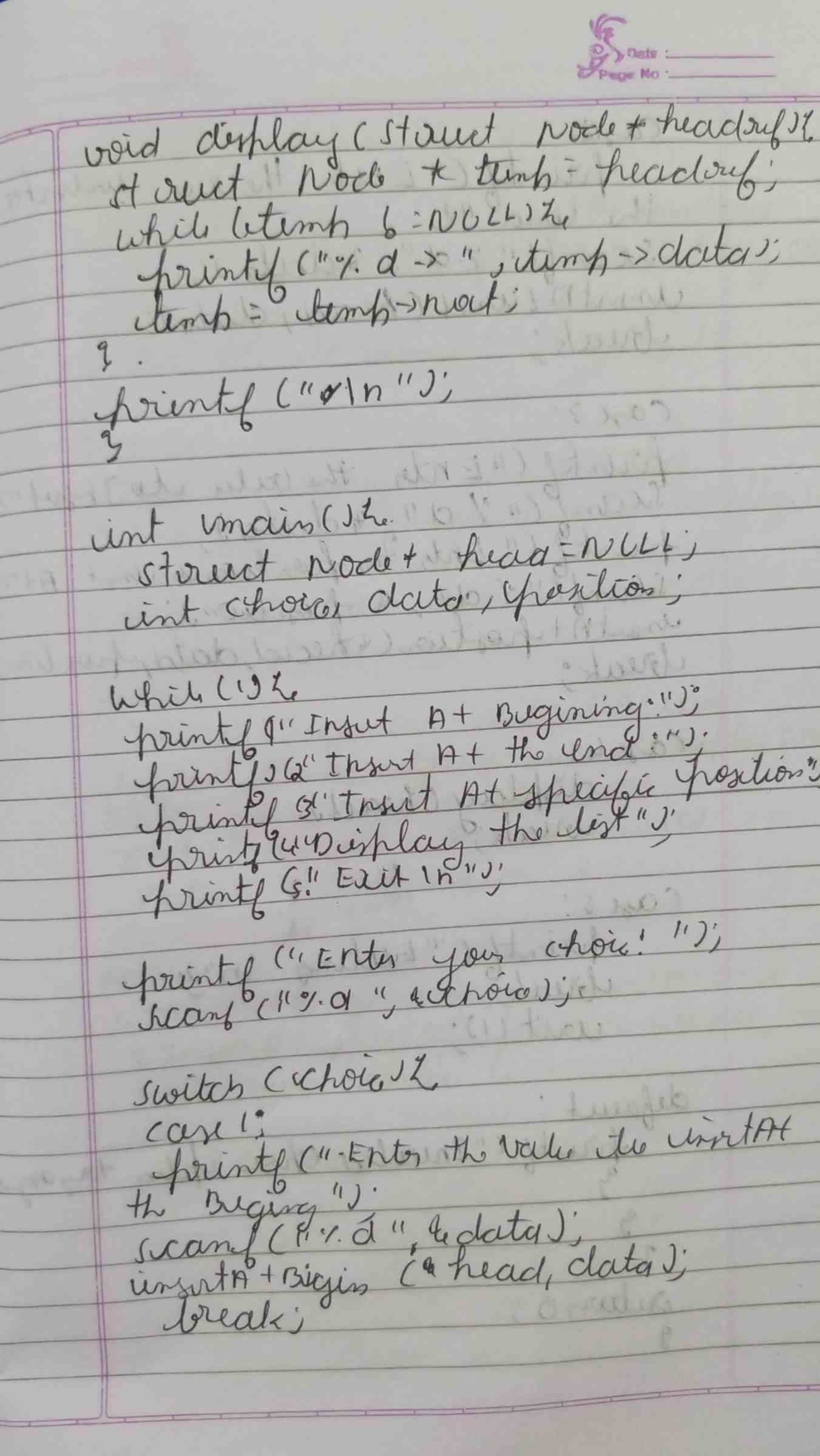
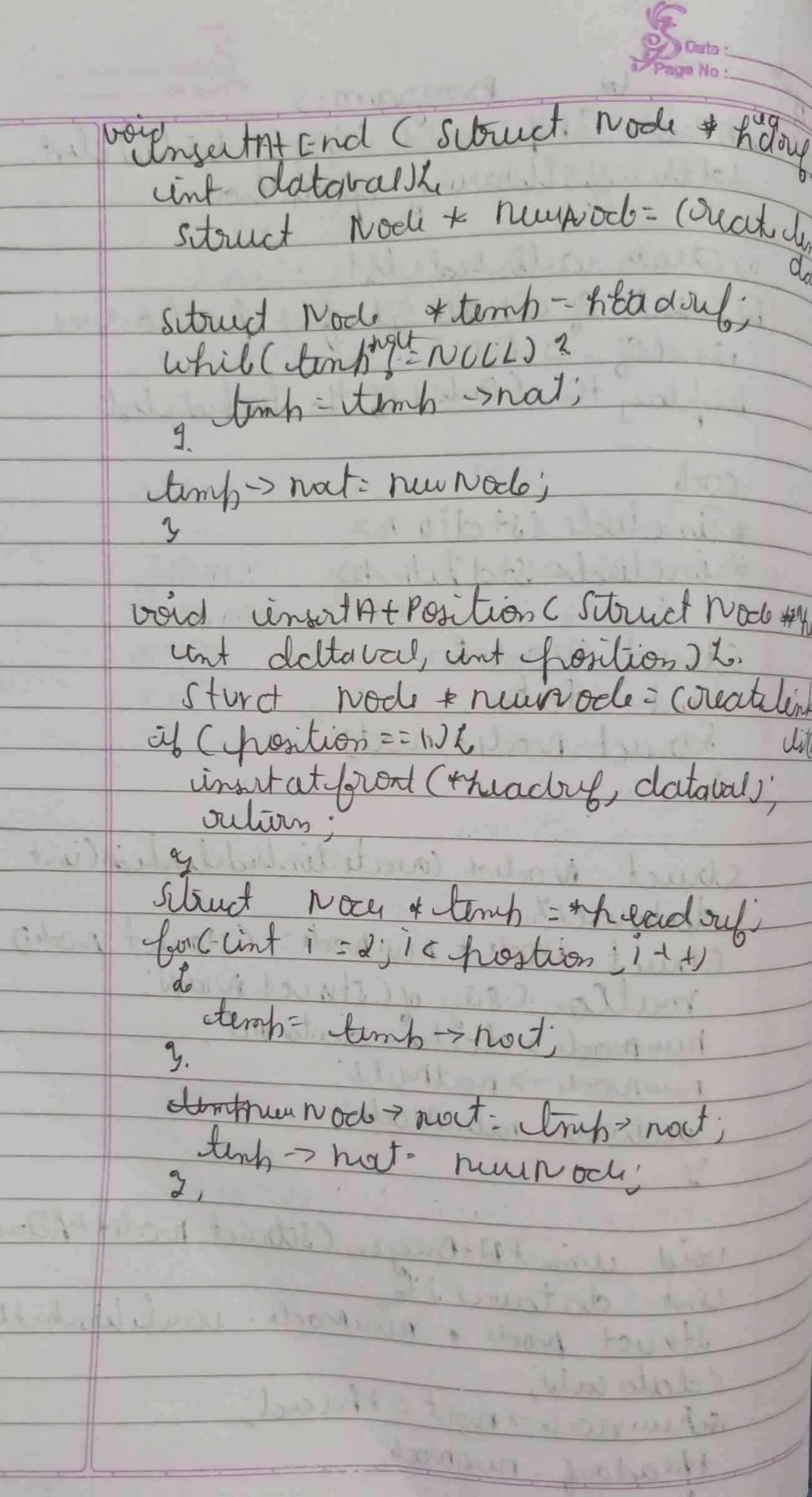
a) Createalinkedlist.

b) Insertion of a node at first position, at any position and at end of list. Display the contents of the linked list.

5b) Leetcode problem no.20 (Valid parantheses)

**OBSERVATION :**

****



**CODE :**

**#include <stdio.h>**

**#include <stdlib.h>**

**struct Node {**

**int data;**

**struct Node\* next;**

**};**

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

**\*head = NULL;**

**}**

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

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

**newNode->data = value;**

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

**\*head = newNode;**

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

**}**

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

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

**newNode->data = value;**

**if (position == 1) {**

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

**\*head = newNode;**

**printf("Node with value %d inserted at position 1.\n", value);**

**return;**

**}**

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

**for (int i = 1; temp != NULL && i < position - 1; i++) {**

**temp = temp->next;**

**}**

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

**printf("Position %d is out of bounds. Insertion failed.\n", position);**

**free(newNode);**

**} else {**

**newNode->next = temp->next;**

**temp->next = newNode;**

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

**}**

**}**

**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;**

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

**return;**

**}**

**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);**

**}**

**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");**

**}**

**int main() {**

**struct Node\* head;**

**createList(&head);**

**int choice, value, position;**

**while (1) {**

**printf("\nLinked List Operations:\n");**

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

**printf("2. Insert at Position\n");**

**printf("3. Insert at End\n");**

**printf("4. Display the list\n");**

**printf("5. Exit\n");**

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

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

**switch (choice) {**

**case 1:**

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

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

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

**break;**

**case 2:**

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

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

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

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

**insertAtPosition(&head, value, position);**

**break;**

**case 3:**

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

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

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

**break;**

**case 4:**

**displayList(head);**

**break;**

**case 5:**

**exit(0);**

**default:**

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

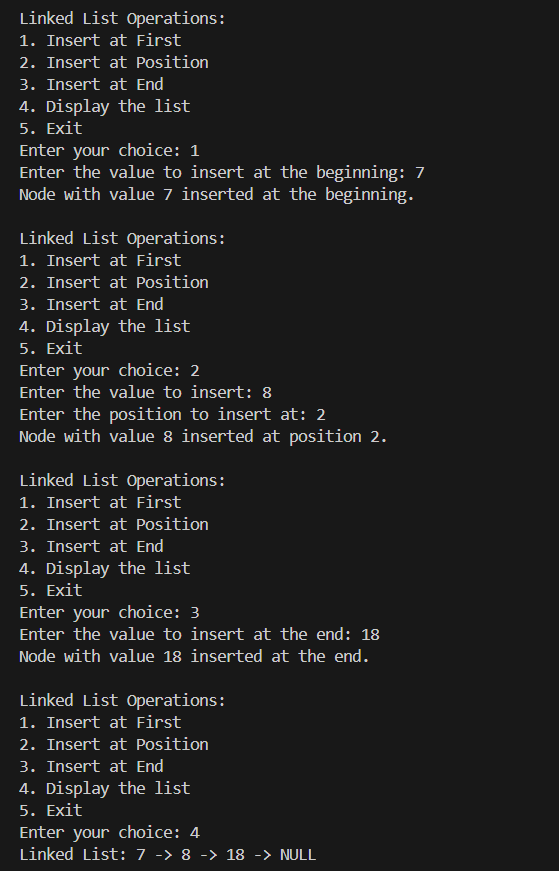
**}**

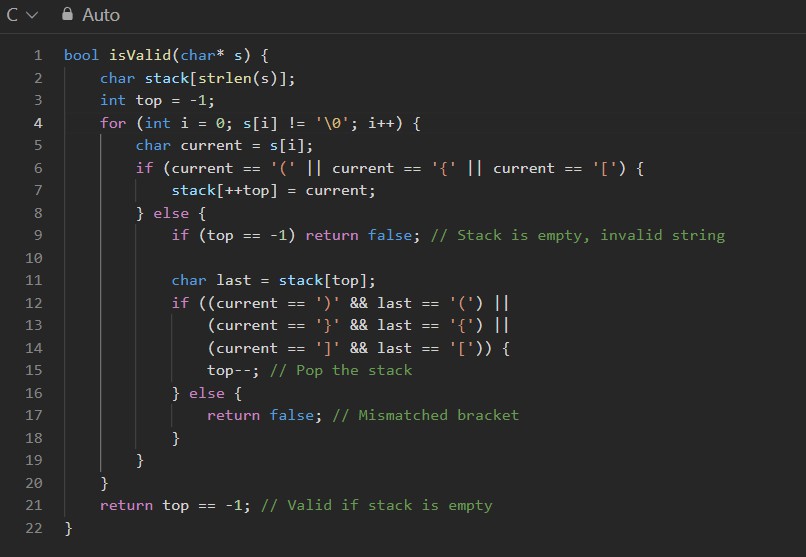
**}**

**return 0;**

**}**

**OUTPUT :**



**LEETCODE PROBLEM :**