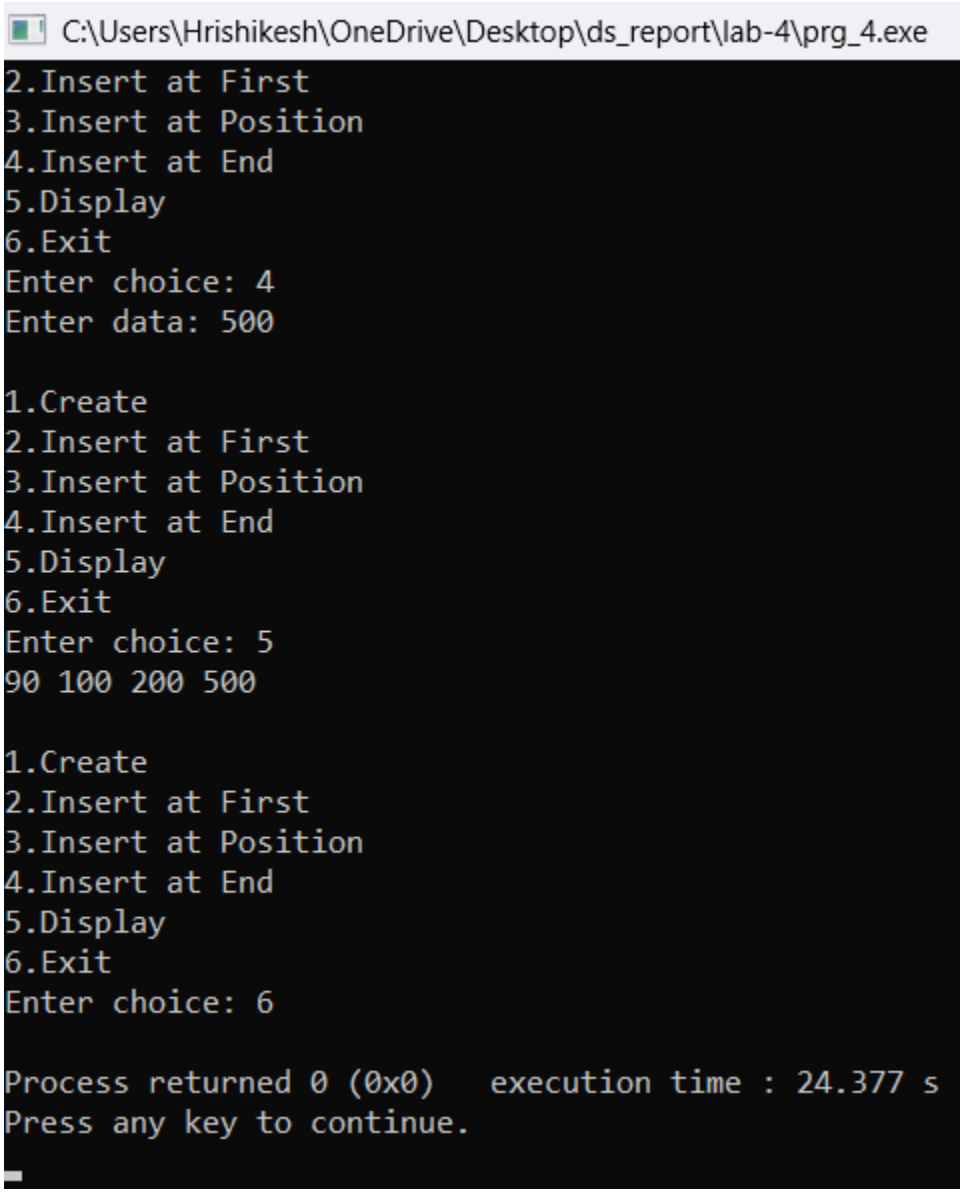
LAB-4 (Singly Linked List-INSERT)

```c
#include <stdio.h>
#include <stdlib.h>

struct node {
    int data;
    struct node *next;
};

struct node *head = NULL;

void create() {
    int n, i, val;
    struct node *temp, *newnode;
    printf("Enter number of nodes: ");
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        newnode = (struct node *)malloc(sizeof(struct node));
        printf("Enter data: ");
        scanf("%d", &val);
        newnode->data = val;
        newnode->next = NULL;
        if (head == NULL) {
            head = newnode;
            temp = head;
        } else {
            temp->next = newnode;
            temp = newnode;
        }
    }
}

void insert_first() {
    int val;
    struct node *newnode;
    newnode = (struct node *)malloc(sizeof(struct node));
    printf("Enter data: ");
    scanf("%d", &val);
    newnode->data = val;
    newnode->next = head;
    head = newnode;
}
```

```c
void insert_end() {
    int val;
    struct node *newnode, *temp;
    newnode = (struct node *)malloc(sizeof(struct node));
    printf("Enter data: ");
    scanf("%d", &val);
    newnode->data = val;
    newnode->next = NULL;
    if (head == NULL) {
        head = newnode;
        return;
    }
    temp = head;
    while (temp->next != NULL)
        temp = temp->next;
    temp->next = newnode;
}

void insert_pos() {
    int pos, val, i;
    struct node *newnode, *temp;
    printf("Enter position: ");
    scanf("%d", &pos);
    printf("Enter data: ");
    scanf("%d", &val);
    if (pos == 1) {
        insert_first();
        return;
    }
    newnode = (struct node *)malloc(sizeof(struct node));
    newnode->data = val;
    temp = head;
    for (i = 1; i < pos - 1 && temp != NULL; i++)
        temp = temp->next;
    if (temp == NULL) {
        printf("Invalid position\n");
        free(newnode);
        return;
```

OUTPUT:


C:\Users\Hrishikesh\OneDrive\Desktop\ds_report\lab-4\prg_4.exe
2.Insert at First
3.Insert at Position
4.Insert at End
5.Display
6.Exit
Enter choice: 4
Enter data: 500
1.Create
2.Insert at First
3.Insert at Position
4.Insert at End
5.Display
6.Exit
Enter choice: 5
90 100 200 500
1.Create
2.Insert at First
3.Insert at Position
4.Insert at End
5.Display
6.Exit
Enter choice: 6
Process returned 0 (0x0) execution time : 24.377 s
Press any key to continue.