WEEK 6:SINGLY LINKED LIST WITH MODIFICATIONS

PROGRAM

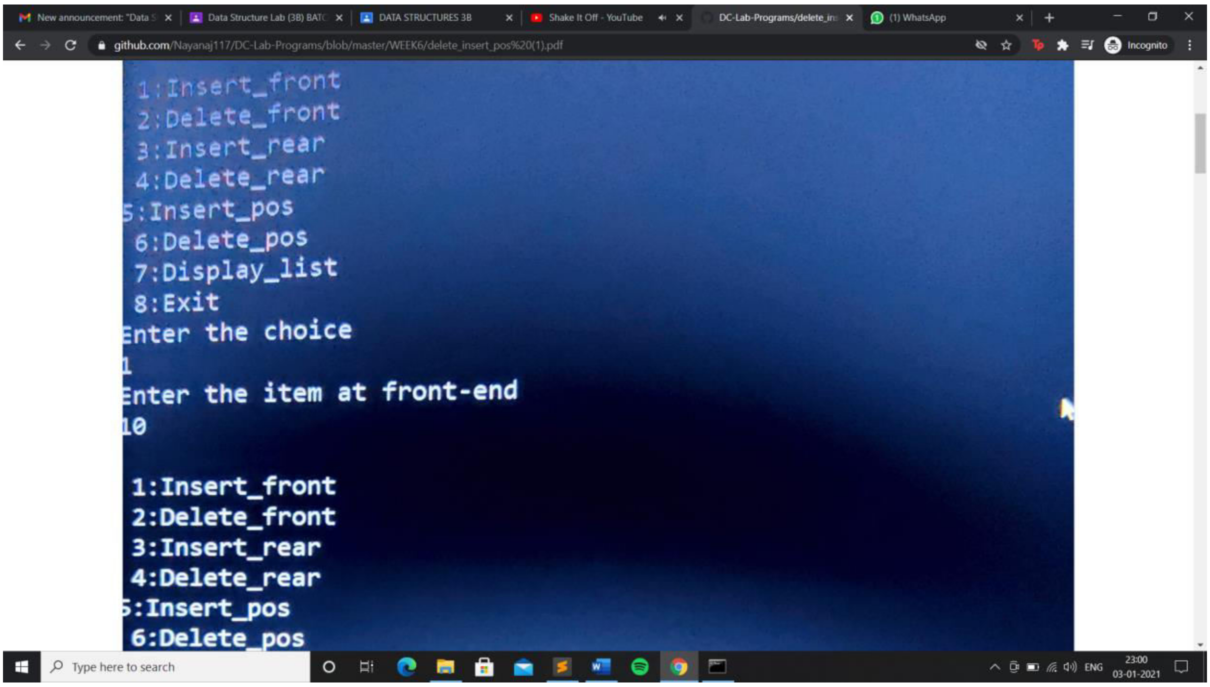
```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
struct node
{
   int info;
   struct node *link;
};
typedef struct node *NODE;
NODE getnode()
{
   NODE x;
   x = (NODE)malloc(sizeof(struct node));
   if (x == NULL)
   {
      printf("mem full\n");
      exit(0);
   }
   return x;
}
void freenode(NODE x)
{
   free(x);
}
NODE insert_front(NODE first, int item)
{
   NODE temp;
   temp = getnode();
```

```
    temp->info = item;
    temp->link = NULL;
    if (first == NULL)
        return temp;
    temp->link = first;
    first = temp;
    return first;
}
NODE delete_front(NODE first)
{
    NODE temp;
    if (first == NULL)
    {
        printf("List is empty cannot delete\n");
        return first;
    }
    temp = first;
    temp = temp->link;
    printf("Item deleted at front-end is=%d\n", first->info);
    free(first);
    return temp;
}
NODE insert_rear(NODE first, int item)
{
    NODE temp, cur;
    temp = getnode();
    temp->info = item;
    temp->link = NULL;
    if (first == NULL)
        return temp;
    cur = first;
    while (cur->link != NULL)
        cur = cur->link;
    cur->link = temp;
    return first;
}
NODE delete_rear(NODE first)
{
    NODE cur, prev;
    if (first == NULL)
    {
        printf("List is empty cannot delete\n");
        return first;
    }
    if (first->link == NULL)
    {
        printf("Item deleted is %d\n", first->info);
        free(first);
        return NULL;
    }
    prev = NULL;
    cur = first;
    while (cur->link != NULL)
    {
        prev = cur;
        cur = cur->link;
    }
    printf("Item deleted at rear-end is %d", cur->info);
    free(cur);
    prev->link = NULL;
    return first;
}
```

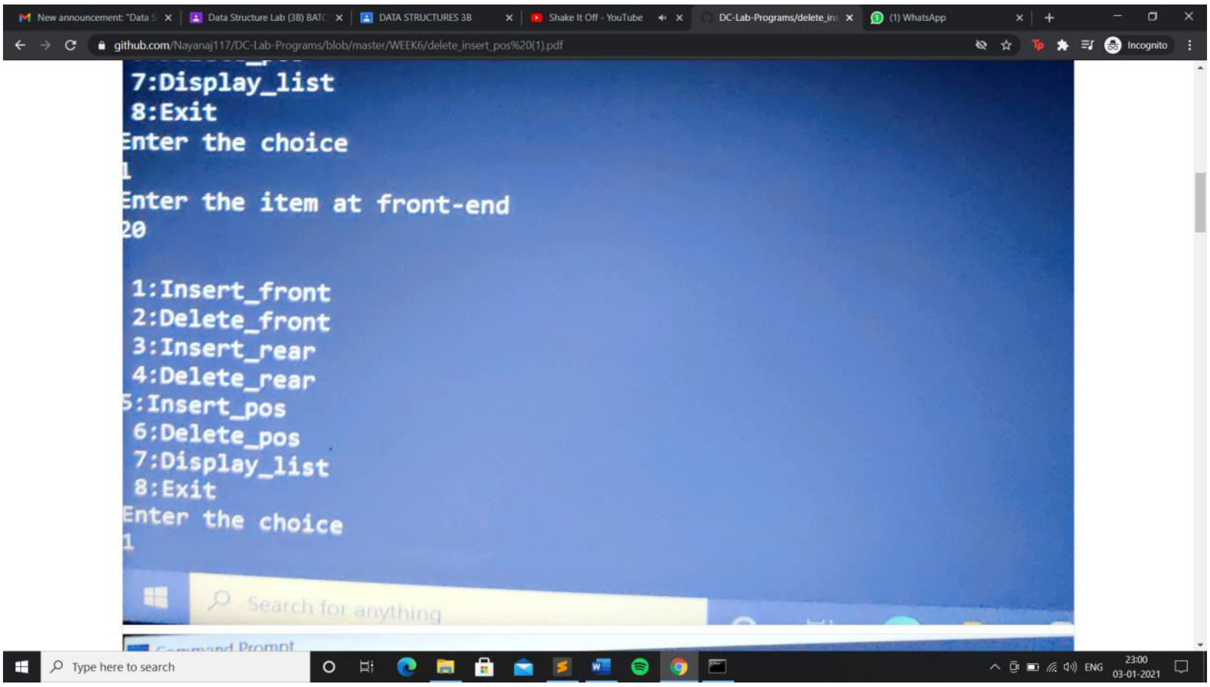
```
NODE insert_pos(int item, int pos, NODE first)
{
  NODE temp;
  NODE prev, cur;
  int count;
  temp = getnode();
  temp->info = item;
  temp->link = NULL;
  if (first == NULL && pos == 1)
    return temp;
  if (first == NULL)
  {
    printf("invalid pos\n");
    return first;
  }
  if (pos == 1)
  {
    temp->link = first;
    return temp;
  }
  count = 1;
  prev = NULL;
  cur = first;
  while (cur != NULL && count != pos)
  {
    prev = cur;
    cur = cur->link;
    count++;
  }
  if (count == pos)
  {
        prev->link = temp;
        temp->link = cur;
        return first;
    }
    printf("IP\n");
    return first;
}

NODE delete_pos(int pos, NODE first)
{
    NODE prev, cur;
    int count;
    if (first == NULL || pos <= 0)
    {
        printf("Invalid position\n");
        return NULL;
    }
    if (pos == 1)
    {
        cur = first;
        first = first->link;
        printf("Item deleted is %d", cur->info);
        freenode(cur);
        return first;
    }
    prev = NULL;
    cur = first;
    count = 1;
    while (cur != NULL)
    {
        if (count == pos)
        {
            break;
        }
        prev = cur;
        cur = cur->link;
        count++;
    }
    if (count != pos)
    {
        printf("Invalid position\n");
        return first;
    }
    prev->link = cur->link;
    printf("Item deleted is %d", cur->info);
    freenode(cur);
    return first;
}
void display(NODE first)
{
    NODE temp;
    if (first == NULL)
        printf("List empty cannot display items\n");
        printf("Contents of the list:\n");
    for (temp = first; temp != NULL; temp = temp->link)
    {
        printf("%d\n", temp->info);
    }
}
void main()
{
    int item, choice, pos;
    NODE first = NULL;

    for (;;)
    {
        printf("\n 1:Insert_front\n 2:Delete_front\n 3:Insert_rear\n 4:Delete_rear\n5:Insert_pos\n 6:Delete_pos\n 7:Display_list\n 8:Exit\n");
        printf("Enter the choice\n");
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
            printf("Enter the item at front-end\n");
            scanf("%d", &item);
            first = insert_front(first, item);
            break;
        case 2:
            first = delete_front(first);
            break;
        case 3:
            printf("Enter the item at rear-end\n");
            scanf("%d", &item);
            first = insert_rear(first, item);
            break;
        case 4:
            first = delete_rear(first);
            break;
        case 5:
        printf("Enter the position and item:\n");
            scanf("%d", &pos);
            scanf("%d",&item);
            first = insert_pos(item, pos, first);
```

```
         break;
      case 6:
         printf("Enter the position:\n");
         scanf("%d", &pos);
         first = delete_pos(pos, first);
         break;
      case 7:
         display(first);
         break;
      case 8:
         exit(0);
         break;
         default:printf("Invalid choice\n");
      }
   }
}
```

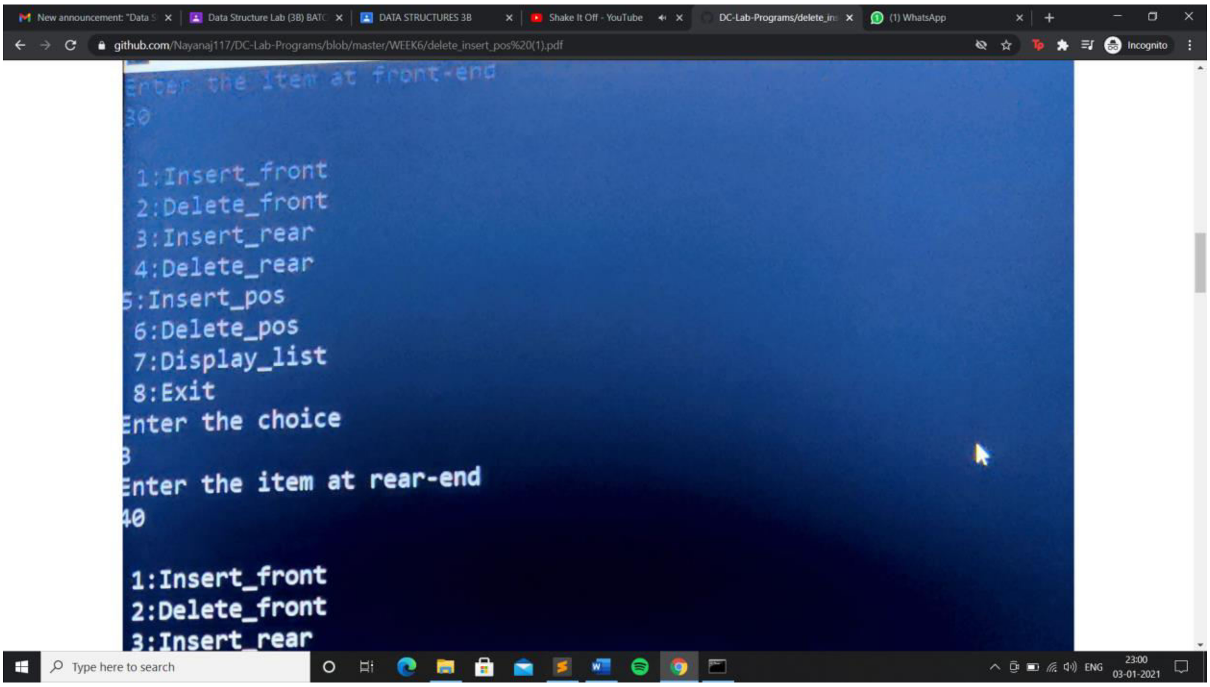
OUTPUT:


New announcement: "Data S
Data Structure Lab (3B) BATC
DATA STRUCTURES 3B
Shake It Off - YouTube
DC-Lab-Programs/delete_ins
(1) WhatsApp
github.com/Nayanaj117/DC-Lab-Programs/blob/master/WEEK6/delete_insert_pos%20(1).pdf
Incognito
Type here to search
ENG
23:00
03-01-2021

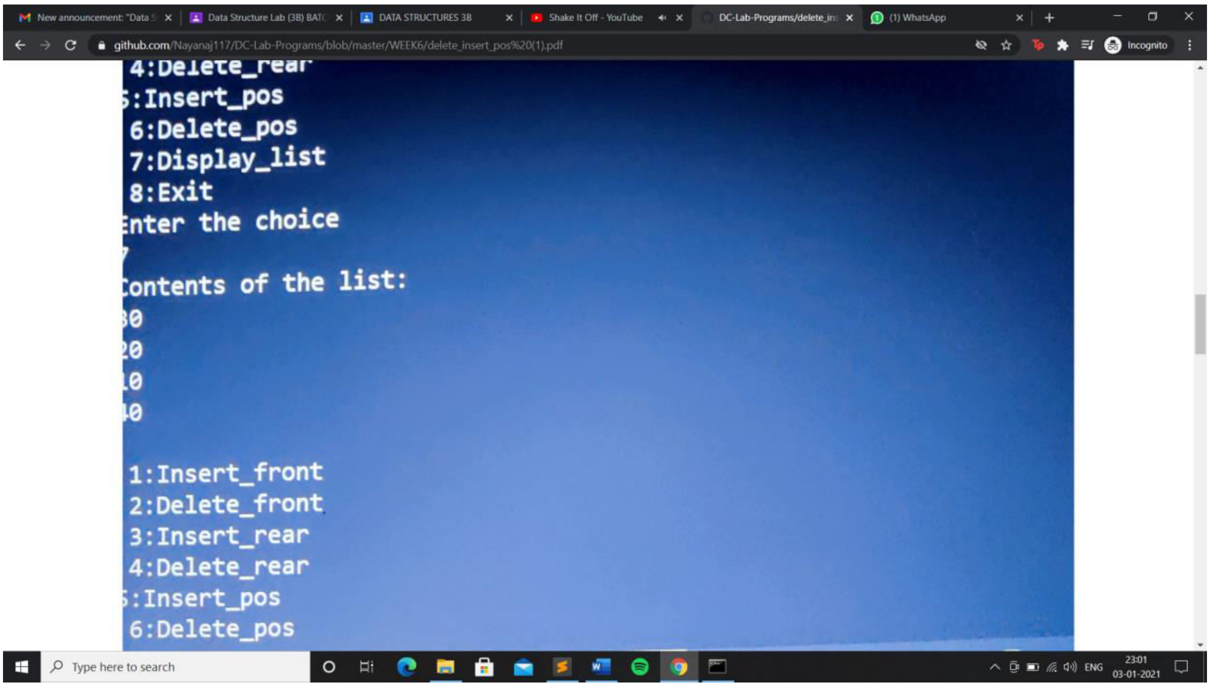
```
 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
5:Insert_pos
 6:Delete_pos
 7:Display_list
 8:Exit
Enter the choice
1
Enter the item at front-end
10

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
5:Insert_pos
 6:Delete_pos
```


New announcement: "Data S
Data Structure Lab (3B) BATC
DATA STRUCTURES 3B
Shake It Off - YouTube
DC-Lab-Programs/delete_in
(1) WhatsApp
github.com/Nayanaj117/DC-Lab-Programs/blob/master/WEEK6/delete_insert_pos%20(1).pdf
Incognito
Search for anything
Command Prompt
Type here to search
ENG
23:00
03-01-2021

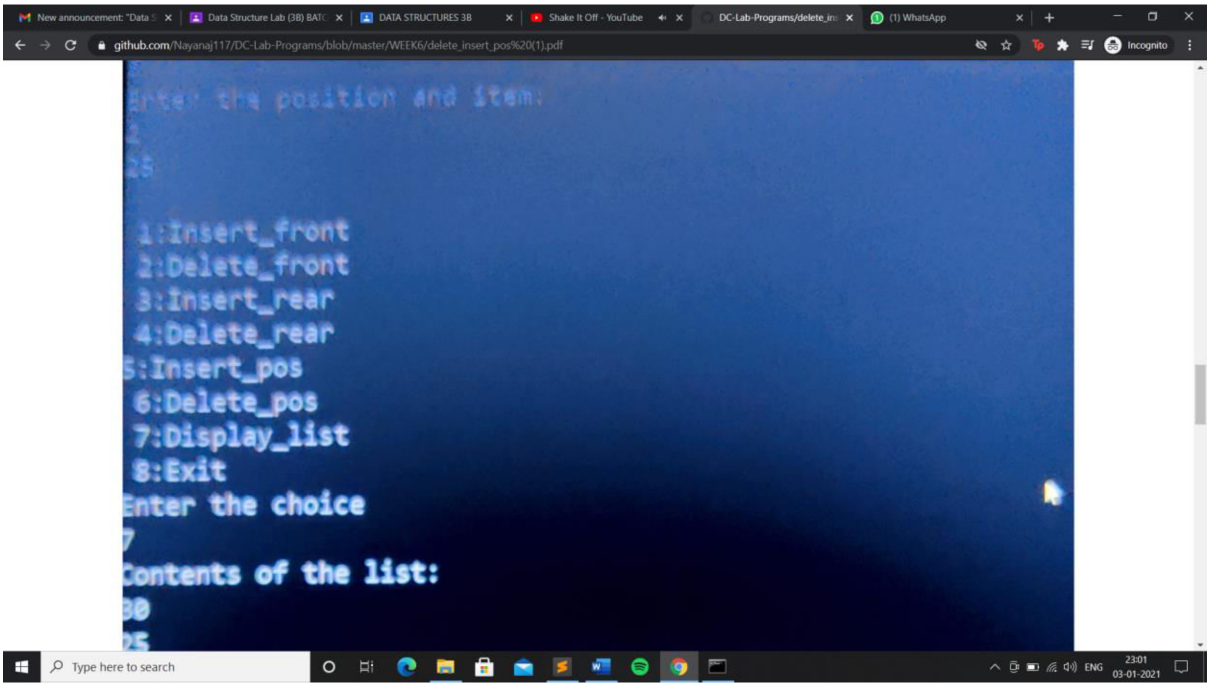
```
 7:Display_list
 8:Exit
Enter the choice
1
Enter the item at front-end
20

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
5:Insert_pos
 6:Delete_pos
 7:Display_list
 8:Exit
Enter the choice
1
```


New announcement: "Data S
Data Structure Lab (3B) BATC
DATA STRUCTURES 3B
Shake It Off - YouTube
DC-Lab-Programs/delete_in
(1) WhatsApp
github.com/Nayanaj117/DC-Lab-Programs/blob/master/WEEK6/delete_insert_pos%20(1).pdf
Incognito
Type here to search
ENG
23:00
03-01-2021

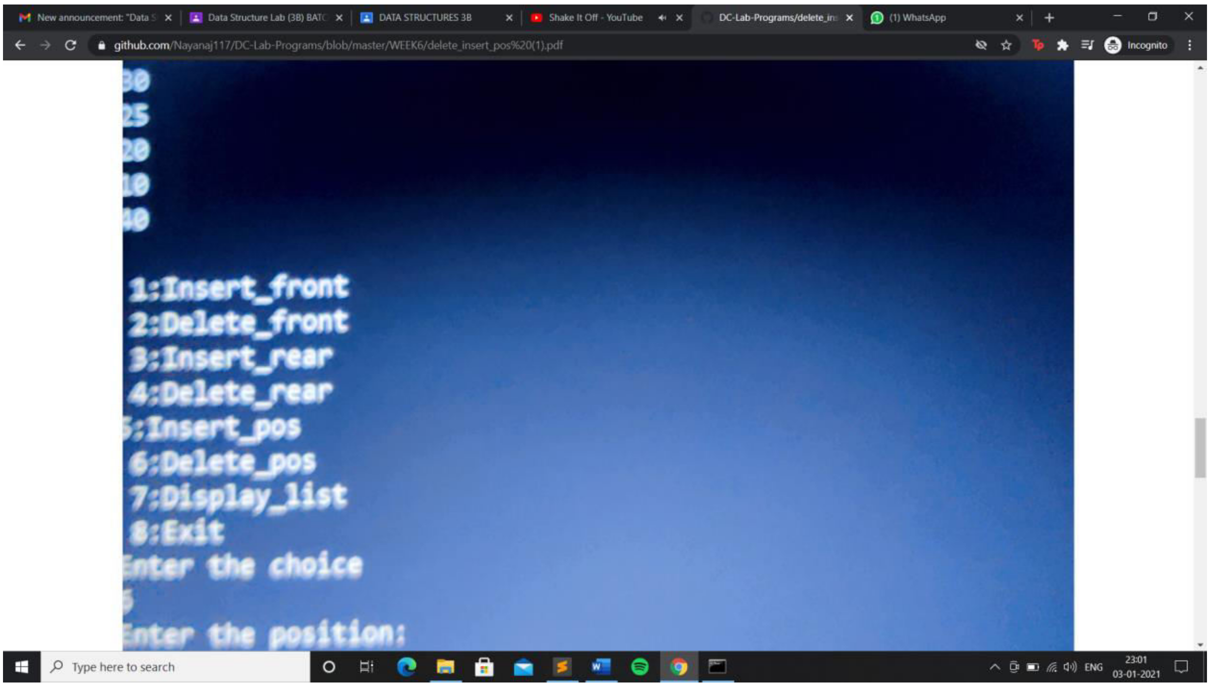
```
Enter the item at front-end
30

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
5:Insert_pos
 6:Delete_pos
 7:Display_list
 8:Exit
Enter the choice
3
Enter the item at rear-end
40

 1:Insert_front
 2:Delete_front
 3:Insert_rear
```


New announcement: "Data S
Data Structure Lab (3B) BATC
DATA STRUCTURES 3B
Shake It Off - YouTube
DC-Lab-Programs/delete_ins
(1) WhatsApp
github.com/Nayanaj117/DC-Lab-Programs/blob/master/WEEK6/delete_insert_pos%20(1).pdf
Incognito
Type here to search
ENG
23:01
03-01-2021

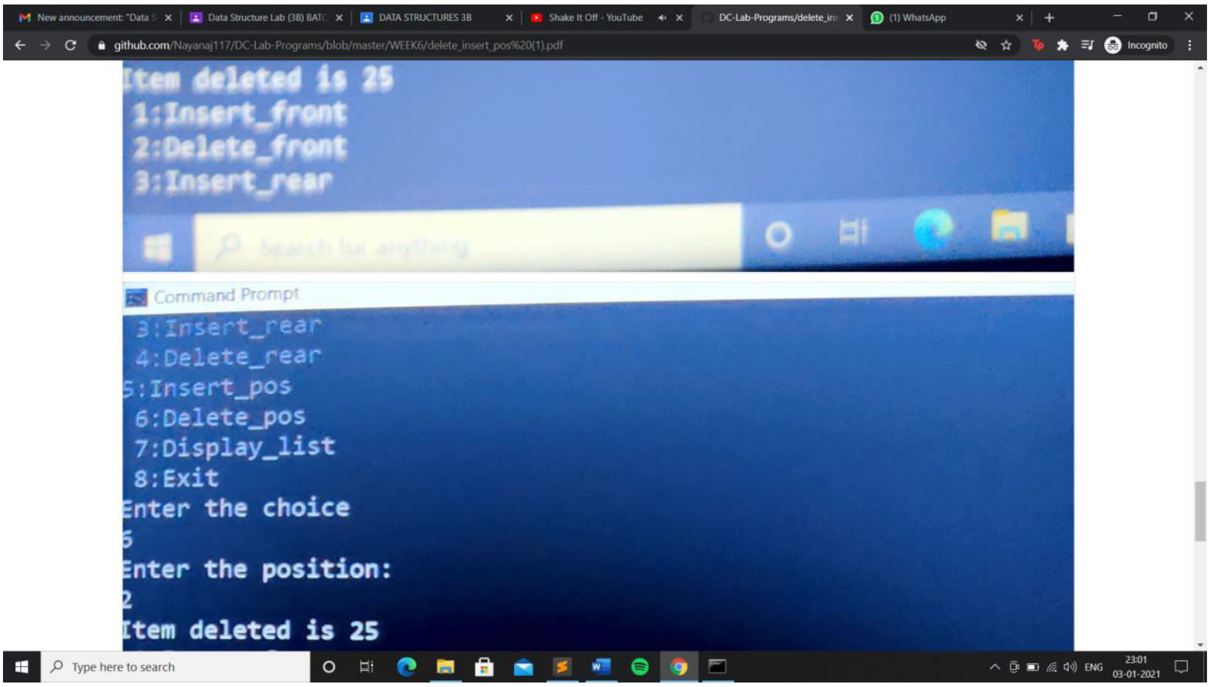
```
 4:Delete_rear
5:Insert_pos
 6:Delete_pos
 7:Display_list
 8:Exit
Enter the choice
7
Contents of the list:
30
20
10
40

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
5:Insert_pos
 6:Delete_pos
```


New announcement: "Data S
Data Structure Lab (3B) BATC
DATA STRUCTURES 3B
Shake It Off - YouTube
DC-Lab-Programs/delete_ins
(1) WhatsApp
github.com/Nayanaj117/DC-Lab-Programs/blob/master/WEEK6/delete_insert_pos%20(1).pdf
Incognito
Type here to search
ENG
23:01
03-01-2021

```
Enter the position and item:
2
25

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
5:Insert_pos
 6:Delete_pos
 7:Display_list
 8:Exit
Enter the choice
7
Contents of the list:
30
25
```

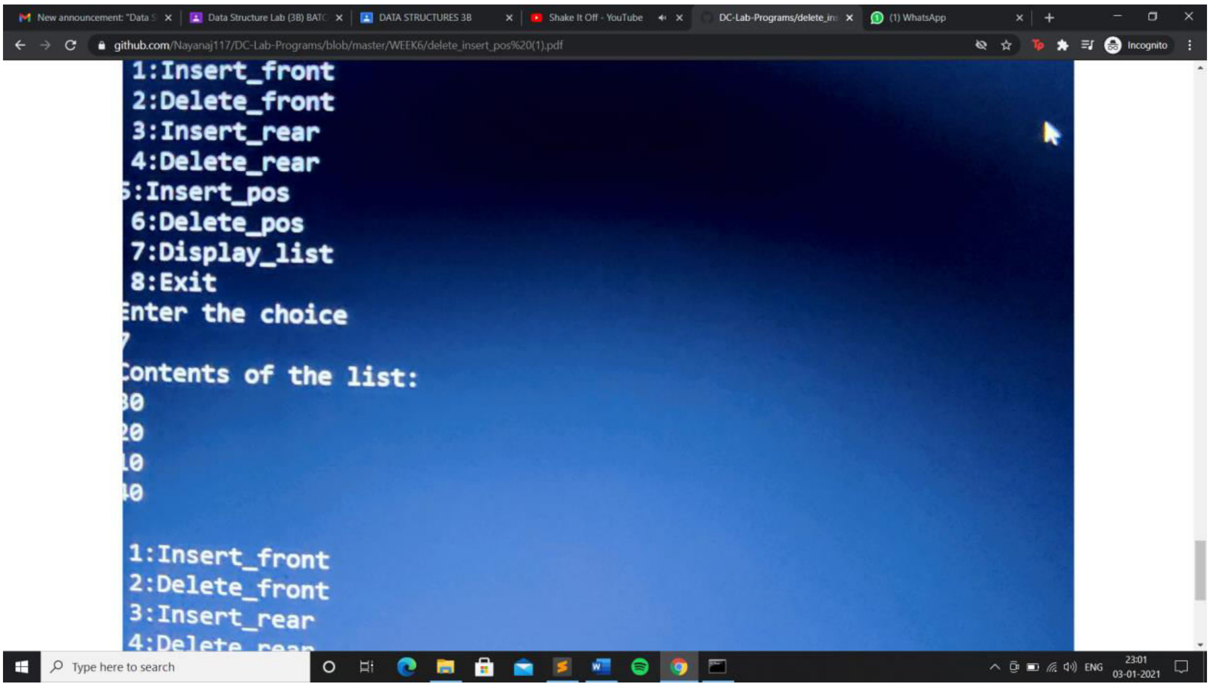

github.com/Nayanaj117/DC-Lab-Programs/blob/master/WEEK6/delete_insert_pos%20(1).pdf
Type here to search

```
30
25
20
10
40


 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
5:Insert_pos
 6:Delete_pos
 7:Display_list
 8:Exit
Enter the choice
6
Enter the position:
```


github.com/Nayanaj117/DC-Lab-Programs/blob/master/WEEK6/delete_insert_pos%20(1).pdf
Command Prompt
Type here to search

```
Item deleted is 25
 1:Insert_front
 2:Delete_front
 3:Insert_rear
```

```
 3:Insert_rear
 4:Delete_rear
5:Insert_pos
 6:Delete_pos
 7:Display_list
 8:Exit
Enter the choice
6
Enter the position:
2
Item deleted is 25
```


New announcement: "Data S
Data Structure Lab (3B) BATC
DATA STRUCTURES 3B
Shake It Off - YouTube
DC-Lab-Programs/delete_ins
(1) WhatsApp
github.com/Nayanaj117/DC-Lab-Programs/blob/master/WEEK6/delete_insert_pos%20(1).pdf
Incognito
Type here to search
ENG
23:01
03-01-2021

```
 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
5:Insert_pos
 6:Delete_pos
 7:Display_list
 8:Exit
Enter the choice
7
Contents of the list:
30
20
10
40

 1:Insert_front
 2:Delete_front
 3:Insert_rear
 4:Delete_rear
```