

WEEK 5:SINGLY LINKED LIST

PROGRAM

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.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\n");
return first;
}
temp=first;
temp=temp->link;
printf("Item deleted at FRONT END=%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\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 %d",cur->info);
    free(cur);
    prev->link=NULL;
    return first;
}

```

```

}

void display(NODE first)
{
    NODE temp;
    if(first==NULL)
        printf("List EMPTY cannot DISPLAY\n");
    for(temp=first;temp!=NULL;temp=temp->link)
    {
        printf("%d\n",temp->info);
    }
}

NODE delete_pos(int pos, NODE first)
{
    NODE cur;
    NODE prev;
    int count;
    if(first==NULL || pos<=0)
    {
        printf("invalid position\n");
        return NULL;
    }
    if (pos==1) similar to key== f->l
    {
        cur=first;
        first=first->link;
        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;
}
if(count!=pos)
{
    printf("invalid position specified\n");
    return first;
}

Pos found so delete the node
prev->link=cur->link;
freenode(cur);
return first;
}

void main()
{
    int item,choice,pos;
    NODE first=NULL;
    for(;;)
    {
        printf("\n1:INSERT FRONT\n2:DELETE FRONT\n3:INSERT REAR\n4:DELETE REAR\n5:DISPLAY\n6:EXIT\n");
    }
}

```

```
printf("Enter the choice:");
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:display(first);
            break;
    default:exit(0);
            break;
}
}
getch();
}
```

OUTPUT:

The screenshot shows a web browser window with the address bar displaying `github.com/Nayanaj117/DC-Lab-Programs/blob/master/WEEK5/singly_list_output%20(1).pdf`. The browser has several tabs open, including "New announcement: 'Data...", "Data Structure Lab (3B) BAT...", "DATA STRUCTURES 3B", "Out Of The Woods - Yo...", "DC-Lab-Programs/singly_list...", and "WhatsApp". The main content area displays a terminal window with a blue background and white text. The terminal shows a menu with six options: 1:INSERT FRONT, 2:DELETE FRONT, 3:INSERT REAR, 4:DELETE REAR, 5:DISPLAY, and 6:EXIT. Below the menu, the prompt "Enter the choice:1" is shown, followed by the prompt "Enter the item at FRONT END:". The user has entered the number "1".

```
1:INSERT FRONT
2:DELETE FRONT
3:INSERT REAR
4:DELETE REAR
5:DISPLAY
6:EXIT
Enter the choice:1
Enter the item at FRONT END:
1
```

The screenshot shows the same web browser window as above, but the terminal window now displays the prompt "Enter the item at FRONT END:" followed by the number "2". The menu options remain the same: 1:INSERT FRONT, 2:DELETE FRONT, 3:INSERT REAR, 4:DELETE REAR, 5:DISPLAY, and 6:EXIT. Below the menu, the prompt "Enter the choice:1" is shown, followed by the prompt "Enter the item at FRONT END:". The user has entered the number "2".

```
Enter the item at FRONT END:
2

1:INSERT FRONT
2:DELETE FRONT
3:INSERT REAR
4:DELETE REAR
5:DISPLAY
6:EXIT
Enter the choice:1
Enter the item at FRONT END:
3
```

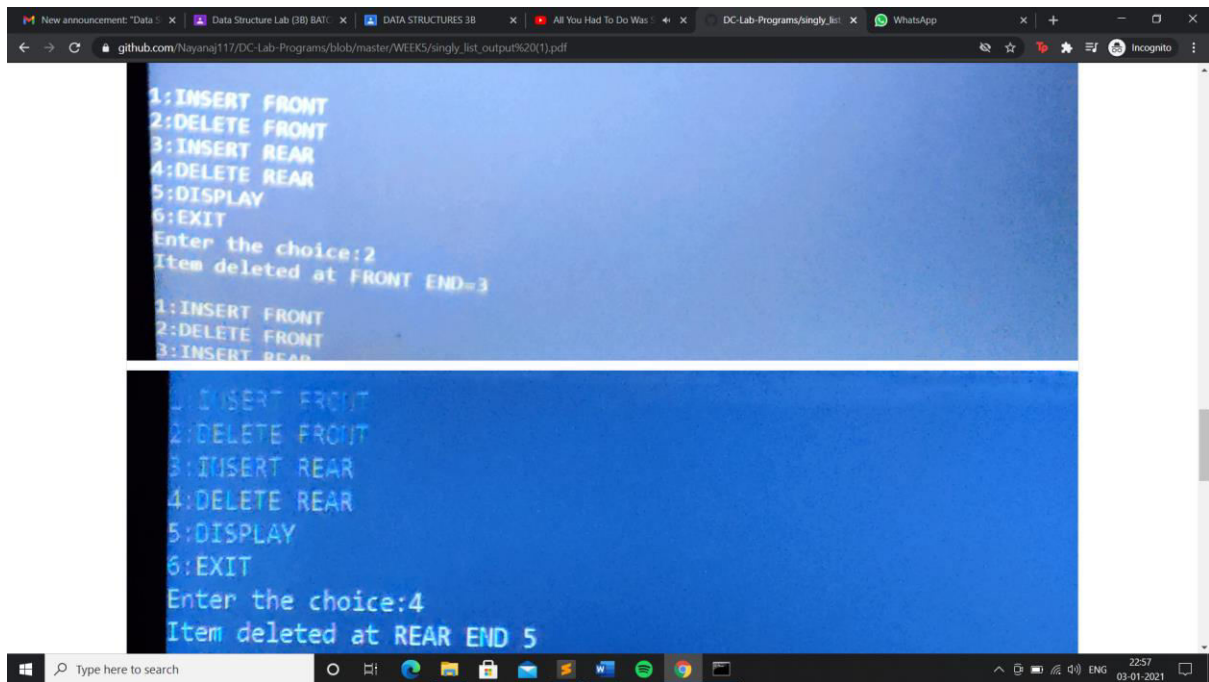
```
Enter the choice:5
3
2
1

1:INSERT FRONT
2:DELETE FRONT
3:INSERT REAR
4:DELETE REAR
5:DISPLAY
6:EXIT
Enter the choice:3
Enter the item at REAR END:
4

1:INSERT FRONT
2:DELETE FRONT
```

```
3:INSERT REAR
4:DELETE REAR
5:DISPLAY
6:EXIT
Enter the choice:3
Enter the item at REAR END:
5

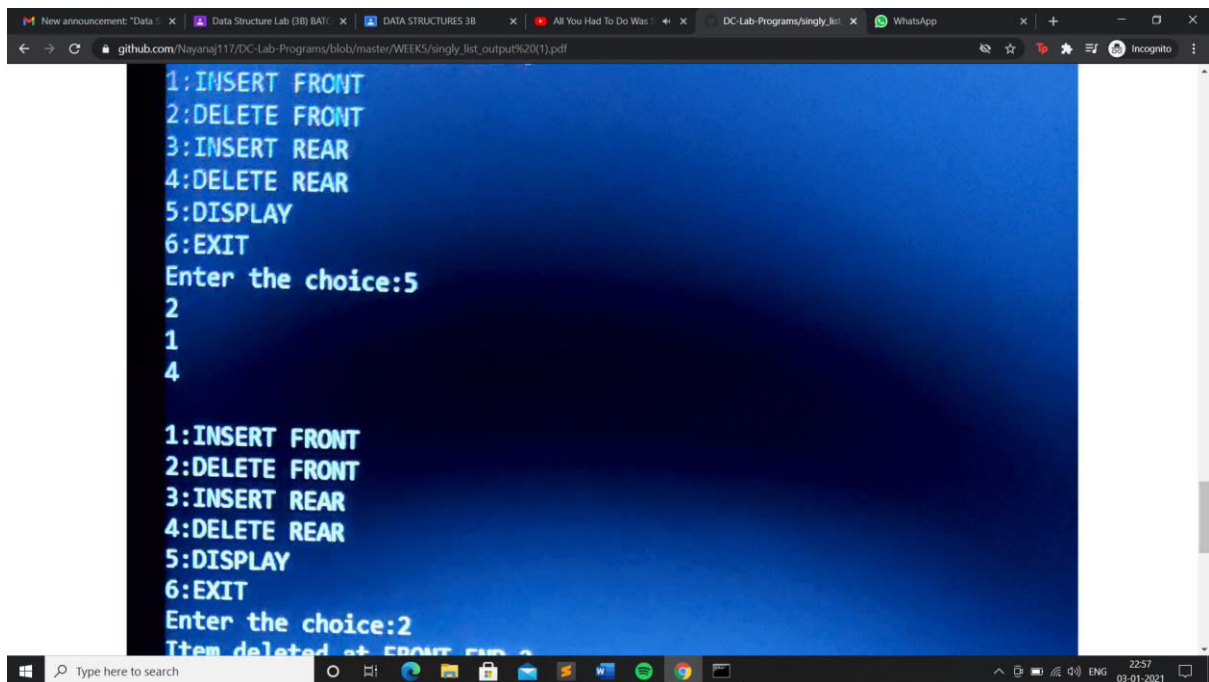
1:INSERT FRONT
2:DELETE FRONT
3:INSERT REAR
4:DELETE REAR
5:DISPLAY
6:EXIT
Enter the choice:5
3
2
1
4
5
```

The screenshot shows a web browser window with multiple tabs. The active tab is a GitHub repository page for 'DC-Lab-Programs/singly_list'. The page content displays two terminal outputs of a program for managing a singly linked list. The first output shows the menu, choice 2, and deletion at the front. The second output shows the menu, choice 4, and deletion at the rear.

```
1:INSERT FRONT
2:DELETE FRONT
3:INSERT REAR
4:DELETE REAR
5:DISPLAY
6:EXIT
Enter the choice:2
Item deleted at FRONT END=3

1:INSERT FRONT
2:DELETE FRONT
3:INSERT REAR
4:DELETE REAR
5:DISPLAY
6:EXIT
Enter the choice:4
Item deleted at REAR END 5
```



The screenshot shows a web browser window with multiple tabs. The active tab is a GitHub repository page for 'DC-Lab-Programs/singly_list'. The page content displays a terminal output of a program for managing a singly linked list. The output shows the menu, choice 5, and the display of the list elements: 2, 1, 4. Below this, the menu is shown again with choice 2, and the start of a deletion operation at the front.

```
1:INSERT FRONT
2:DELETE FRONT
3:INSERT REAR
4:DELETE REAR
5:DISPLAY
6:EXIT
Enter the choice:5
2
1
4

1:INSERT FRONT
2:DELETE FRONT
3:INSERT REAR
4:DELETE REAR
5:DISPLAY
6:EXIT
Enter the choice:2
Item deleted at FRONT END=3
```

