Institute of Computer Technology

B. Tech Computer Science and Engineering

Subject: DS (2CSE302)

# PRACTICAL-18

**AIM: - Implement the scenario based on doubly linked list & circular doubly linked list.**

***Ques. Jay wants to add forward and backward button in his website which directs the addresses to previous and next from the current value. Refer the given scenario and provide appropriate solution using doubly linked list:***

**1. Insert new item at the first position**
**2. Insert new item at the last position**
**3. Insert new item before the given value of node**
**4. Insert new item after the given value of node**
**5. Delete item from the first position**
**6. Delete item from the last position**
**7. Delete particular item**
**8. Delete item before the given item**
**9. Delete item after the given item**
**10. Search item from the list**
**11. Display all elements of the list**

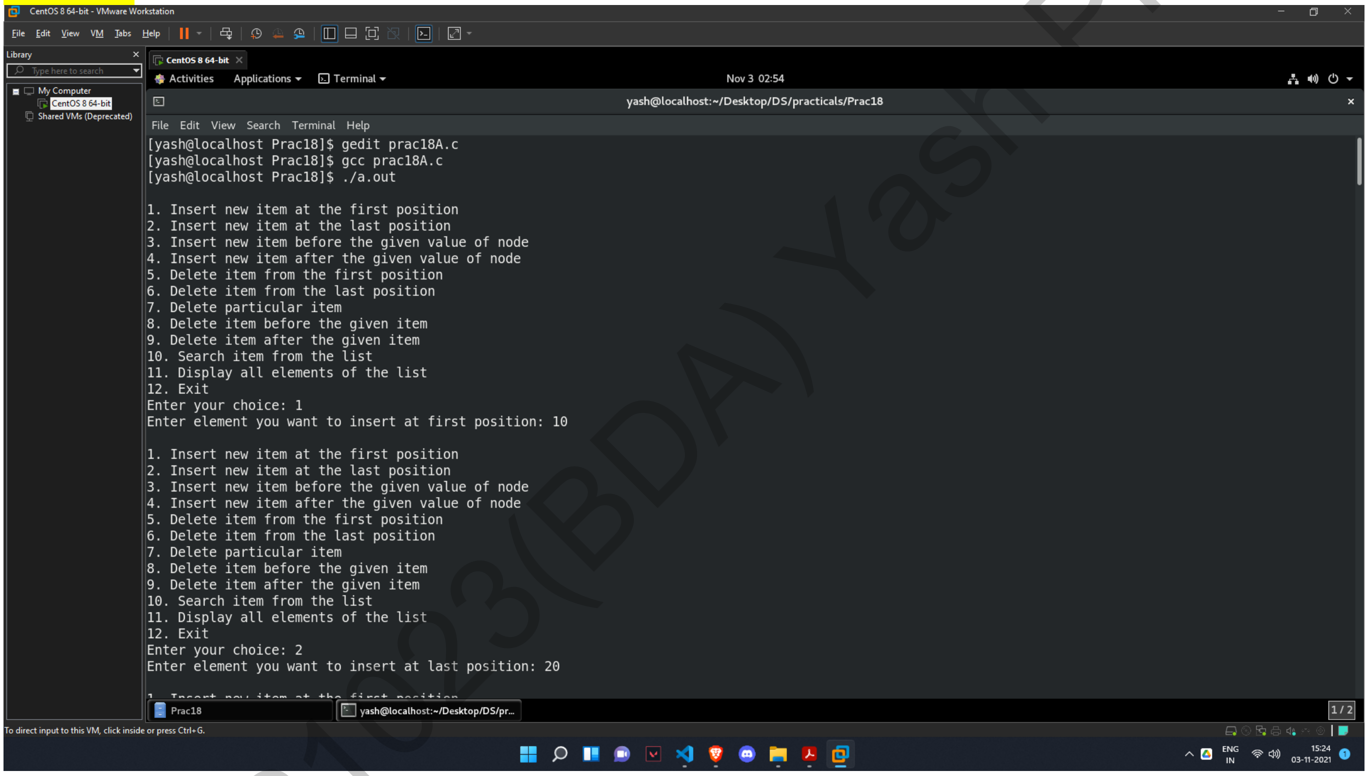
***SOLUTION***

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
        int data;
        struct node *previous, *next;
};
struct node *start;
void insert_begin()
{
        int yash;
        printf("Enter element you want to insert at first position: ");
        scanf("%d", &yash);
        struct node *new_node;
        new_node = (struct node *)malloc(sizeof(struct node));
        new_node->data = yash;
        if (start == NULL)
        {
                start = new_node;
                new_node->next = NULL;
                new_node->previous = NULL;
        }
        else
        {
                start->previous = new_node;
                new_node->next = start;
                new_node->previous = NULL;
                start = new_node;
        }
}
void insert_end()
{
        int yash;
        printf("Enter element you want to insert at last position: ");
        scanf("%d", &yash);
        struct node *new_node, *ptr;
        new_node = (struct node *)malloc(sizeof(struct node));
        ptr = start;
        new_node->data = yash;
        new_node->next = NULL;
        if (start == NULL)
        {
                start = new_node;
                new_node->previous = NULL;
        }
        else
        {
                while (ptr->next != NULL)
                {
                        ptr = ptr->next;
                }
                new_node->previous = ptr;
                ptr->next = new_node;
        }
}
void insert_before()
{
        int yash, item;
        printf("Enter element you want to insert: ");
        scanf("%d", &yash);
        printf("Enter element before which yo want to insert: ");
        scanf("%d", &item);
        struct node *new_node, *ptr, *before;
        new_node = (struct node *)malloc(sizeof(struct node));
        new_node->data = yash;
        ptr = start;
        before = ptr;
        while (ptr->data != item)
        {
                before = ptr;
                ptr = ptr->next;
        }
        before->next = new_node;
        new_node->previous = before;
        new_node->next = ptr;
        ptr->previous = new_node;
}
void insert_after()
{
        int yash, item;
        printf("Enter element you want to insert: ");
        scanf("%d", &yash);
        printf("Enter element after which yo want to insert: ");
        scanf("%d", &item);
        struct node *new_node, *ptr;
        new_node = (struct node *)malloc(sizeof(struct node));
        new_node->data = yash;
        ptr = start;
        while (ptr->data != item)
        {
                ptr = ptr->next;
        }
        new_node->next = ptr->next;
        new_node->previous = ptr;
        ptr->next = new_node;
}
void delete_begin()
{
        struct node *ptr;
        ptr = start;
        start = start->next;
        start->previous = NULL;
        free(ptr);
}
void delete_last()
{
        struct node *ptr;
        ptr = start;
        while (ptr->next != NULL)
        {
                ptr = ptr->next;
        }
        ptr->previous->next = NULL;
        free(ptr);
}
void delete ()
{
        int yash;
        printf("Enter element which you want to delete: ");
        scanf("%d", &yash);
        struct node *ptr;
        ptr = start;
        while (ptr->data != yash)
        {
                ptr = ptr->next;
        }
        ptr->previous->next = ptr->next;
        ptr->next->previous = ptr->previous;
        free(ptr);
}
void delete_before()
{
        int yash;
        printf("Enter element before which you want to delete: ");
        scanf("%d", &yash);
        struct node *ptr, *before;
        ptr = start;
        before = ptr;
        while (ptr->data != yash)
        {
                before = ptr;
                ptr = ptr->next;
        }
        ptr->previous = ptr->previous->previous;
        before->previous->next = ptr;
        free(before);
}
void delete_after()
{
        int yash;
        printf("Enter element after which you want to delete: ");
        scanf("%d", &yash);
```

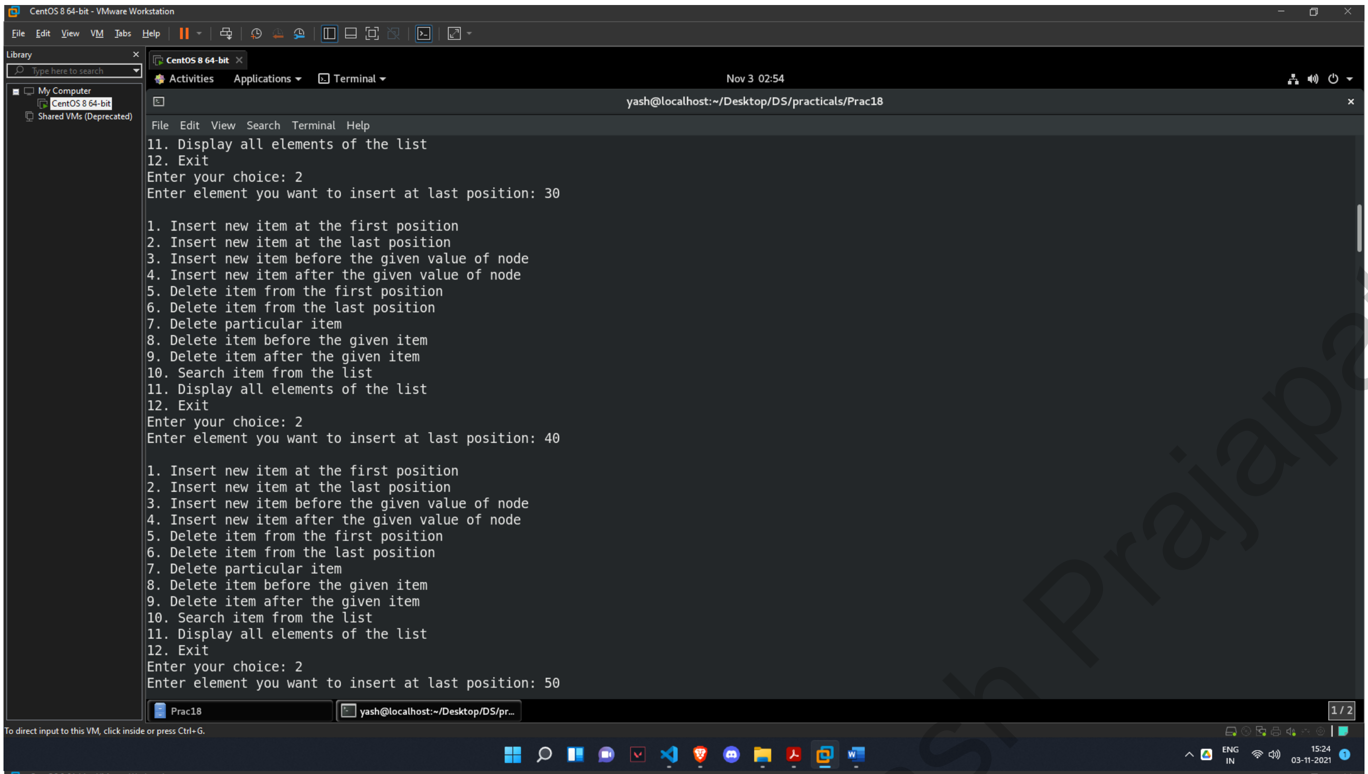
```
		struct node *ptr, *after;
		ptr = start;
		while (ptr->data != yash)
		{
			ptr = ptr->next;
		}
		after = ptr->next;
		ptr->next = after->next;
		after->next->previous = after->previous;
		free(after);
	}
	void search()
	{
		int yash, pos = 0;
		printf("Insert item which you want to search: ");
		scanf("%d", &yash);
		struct node *ptr;
		ptr = start;
		while (ptr->next != NULL)
		{
			if (ptr->data == yash)
			{
				break;
			}
			else
			{
				ptr = ptr->next;
				pos++;
			}
		}
		printf("Item is available at the %d position of the list\n", pos + 1);
	}
	void display()
	{
		printf("List is: \n");
		struct node *ptr;
		ptr = start;
		while (ptr != NULL)
		{
			printf("%d\n", ptr->data);
			ptr = ptr->next;
		}
	}
	int main()
```

```
{
        int choice;
        for (;;)
        {
                printf("\n1. Insert new item at the first position\n2. Insert new item at the last position\n3. Insert new item before the given value of node\n4. Insert new item after the given value of node\n5. Delete item from the first position\n6. Delete item from the last position\n7. Delete particular item\n8. Delete item before the given item\n9. Delete item after the given item\n10. Search item from the list\n11. Display all elements of the list\n12. Exit\n");
                printf("Enter your choice: ");
                scanf("%d", &choice);
                switch (choice)
                {
                case 1:
                        insert_begin();
                        break;
                case 2:
                        insert_end();
                        break;
                case 3:
                        insert_before();
                        break;
                case 4:
                        insert_after();
                        break;
                case 5:
                        delete_begin();
                        break;
                case 6:
                        delete_last();
                        break;
                case 7:
                        delete ();
                        break;
                case 8:
                        delete_before();
                        break;
                case 9:
                        delete_after();
                        break;
                case 10:

                        search();
                        break;
                case 11:
                        display();
                        break;
                case 12:
                        exit(0);
                        break;
                }
        }
        return 0;
}
```

***OUTPUT***

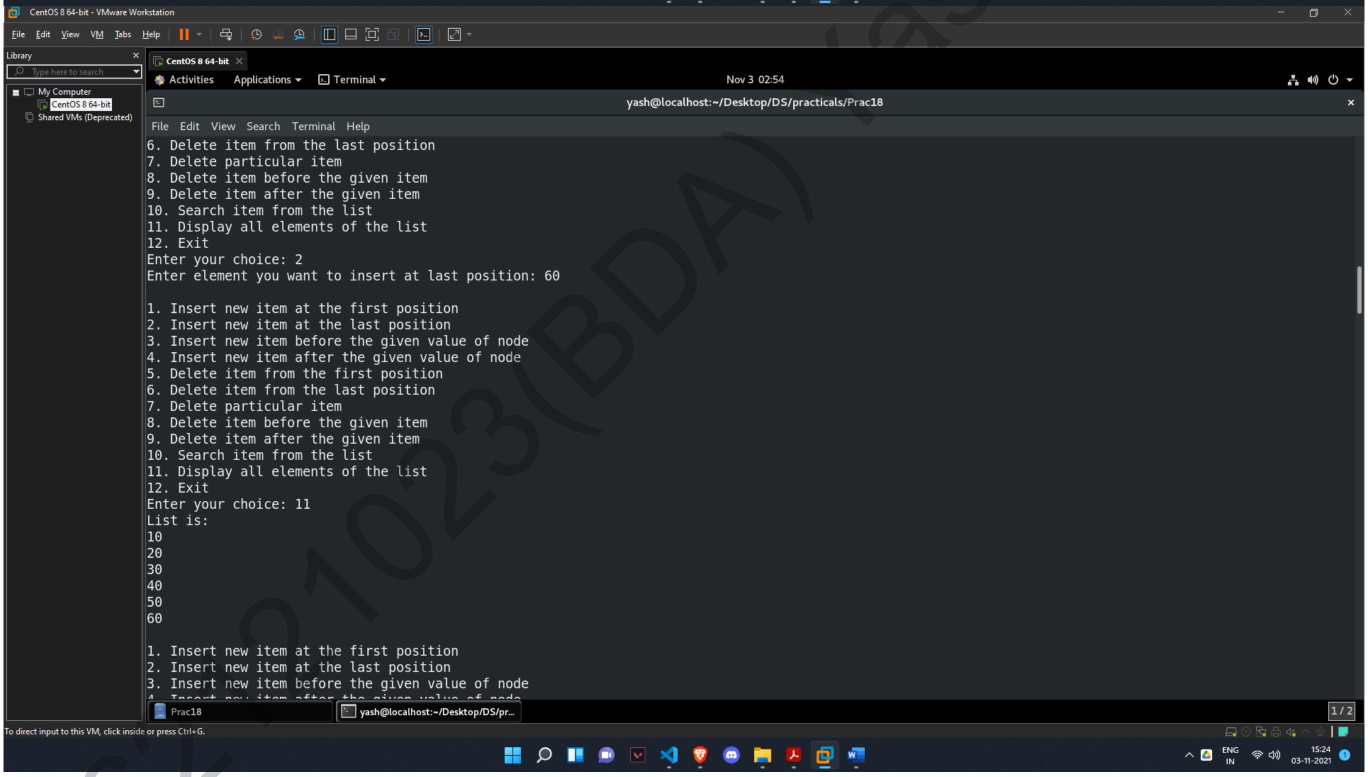

CentOS 8 64-bit - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
CentOS 8 64-bit
Shared VMs (Deprecated)
Activities
Applications
Terminal
Nov 3 02:54
yash@localhost:~/Desktop/DS/practicals/Prac18
File Edit View Search Terminal Help
Prac18
yash@localhost:~/Desktop/DS/pr...
1 / 2
To direct input to this VM, click inside or press Ctrl+G.
ENG IN
15:24
03-11-2021

```
[yash@localhost Prac18]$ gedit prac18A.c
[yash@localhost Prac18]$ gcc prac18A.c
[yash@localhost Prac18]$ ./a.out

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 1
Enter element you want to insert at first position: 10

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 2
Enter element you want to insert at last position: 20

1. Insert new item at the first position
```



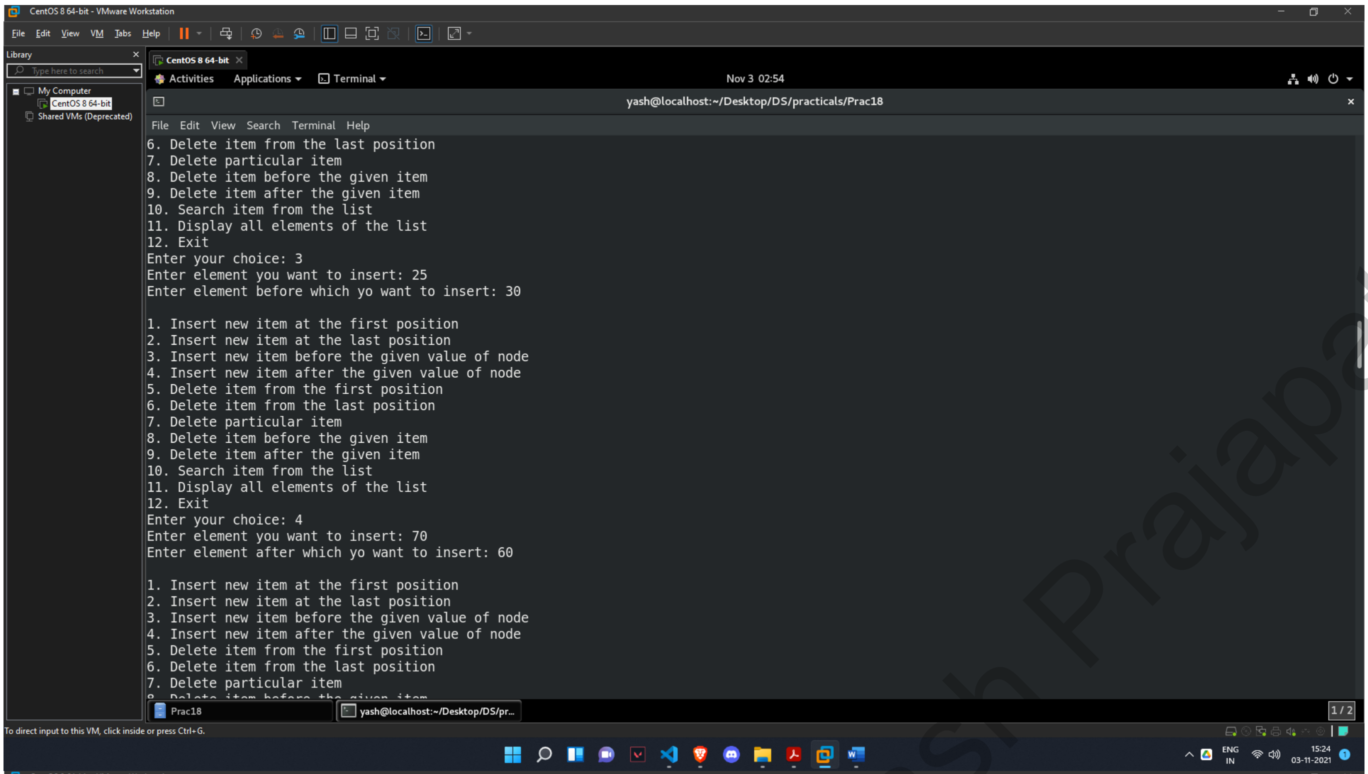
CentOS 8 64-bit - VMware Workstation
Activities
Applications
Terminal
Nov 3 02:54
yash@localhost:~/Desktop/DS/practicals/Prac18
Prac18

```
11. Display all elements of the list
12. Exit
Enter your choice: 2
Enter element you want to insert at last position: 30

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 2
Enter element you want to insert at last position: 40

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 2
Enter element you want to insert at last position: 50
```

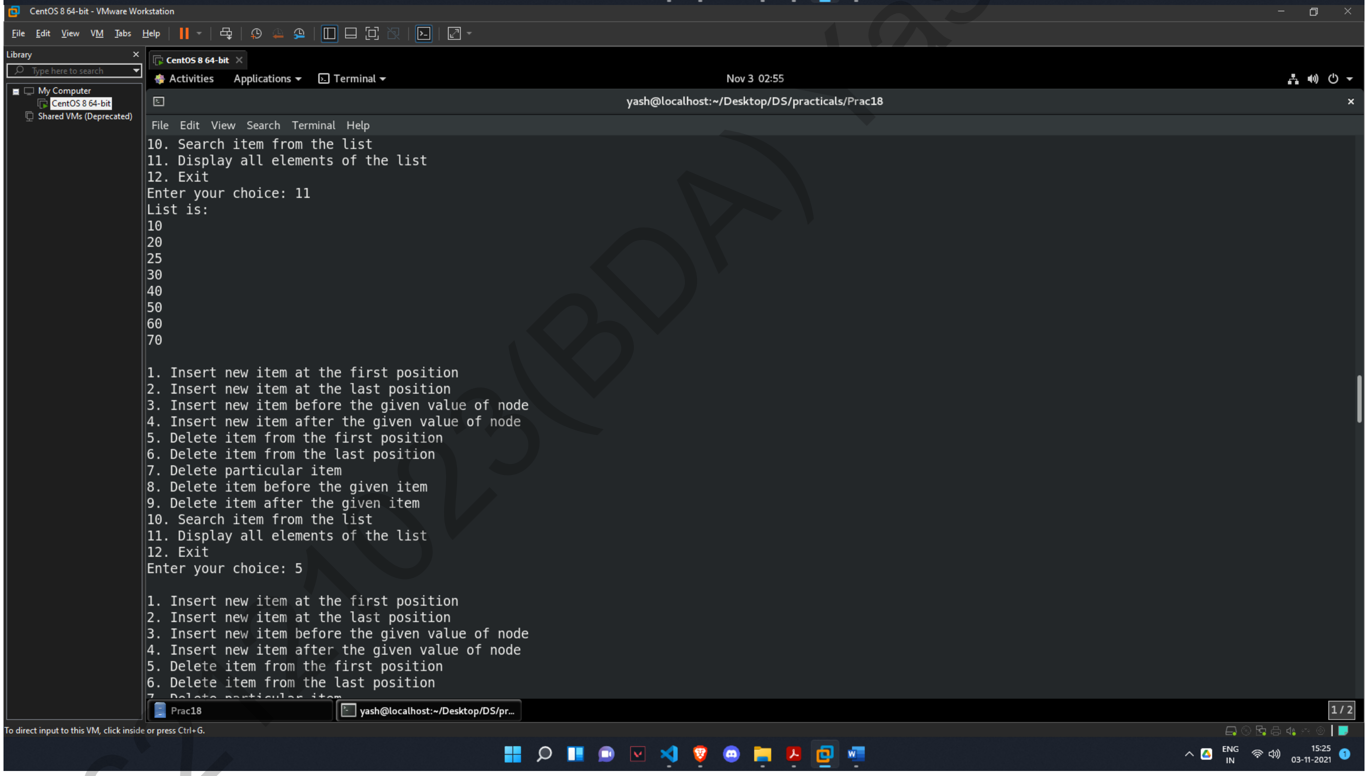


CentOS 8 64-bit - VMware Workstation
Activities
Applications
Terminal
Nov 3 02:54
yash@localhost:~/Desktop/DS/practicals/Prac18
Prac18

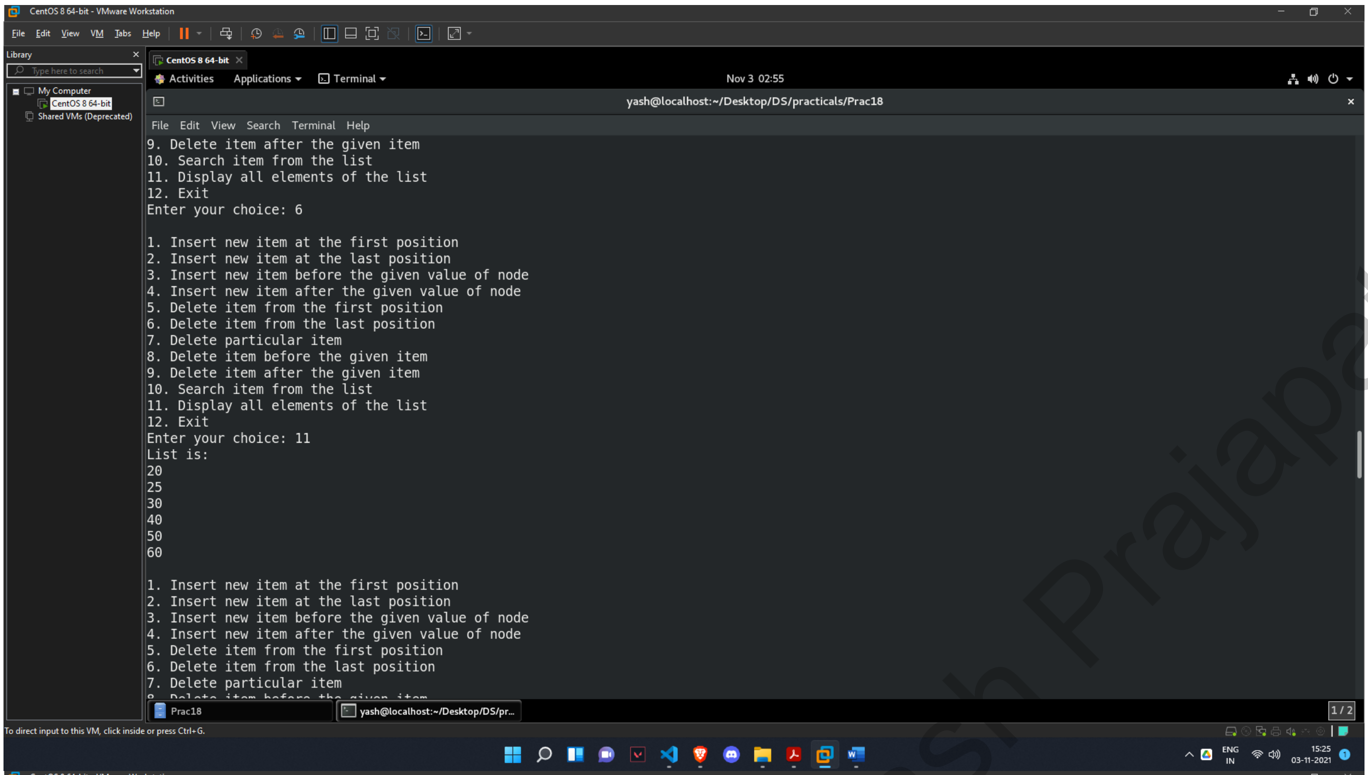
```
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 2
Enter element you want to insert at last position: 60

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 11
List is:
10
20
30
40
50
60

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
```



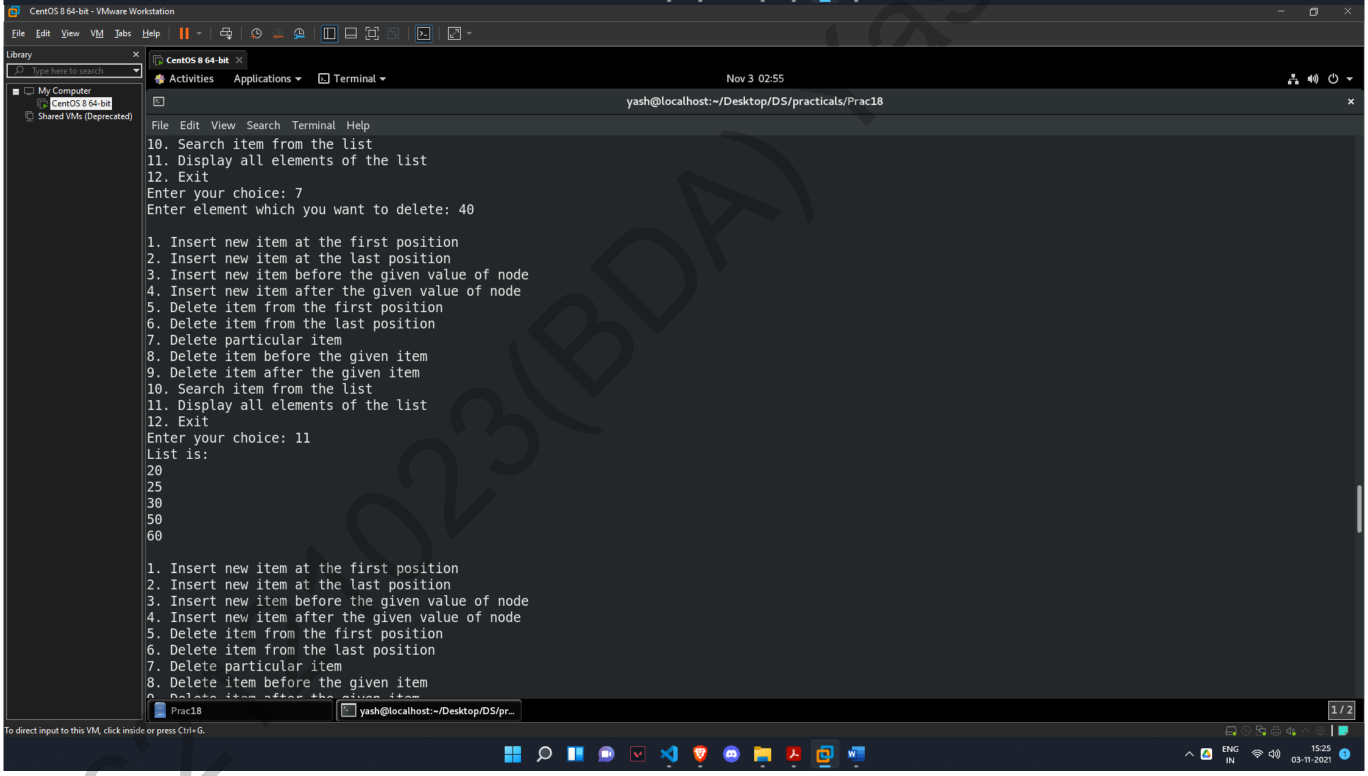
```
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 3
Enter element you want to insert: 25
Enter element before which yo want to insert: 30

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 4
Enter element you want to insert: 70
Enter element after which yo want to insert: 60

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
```



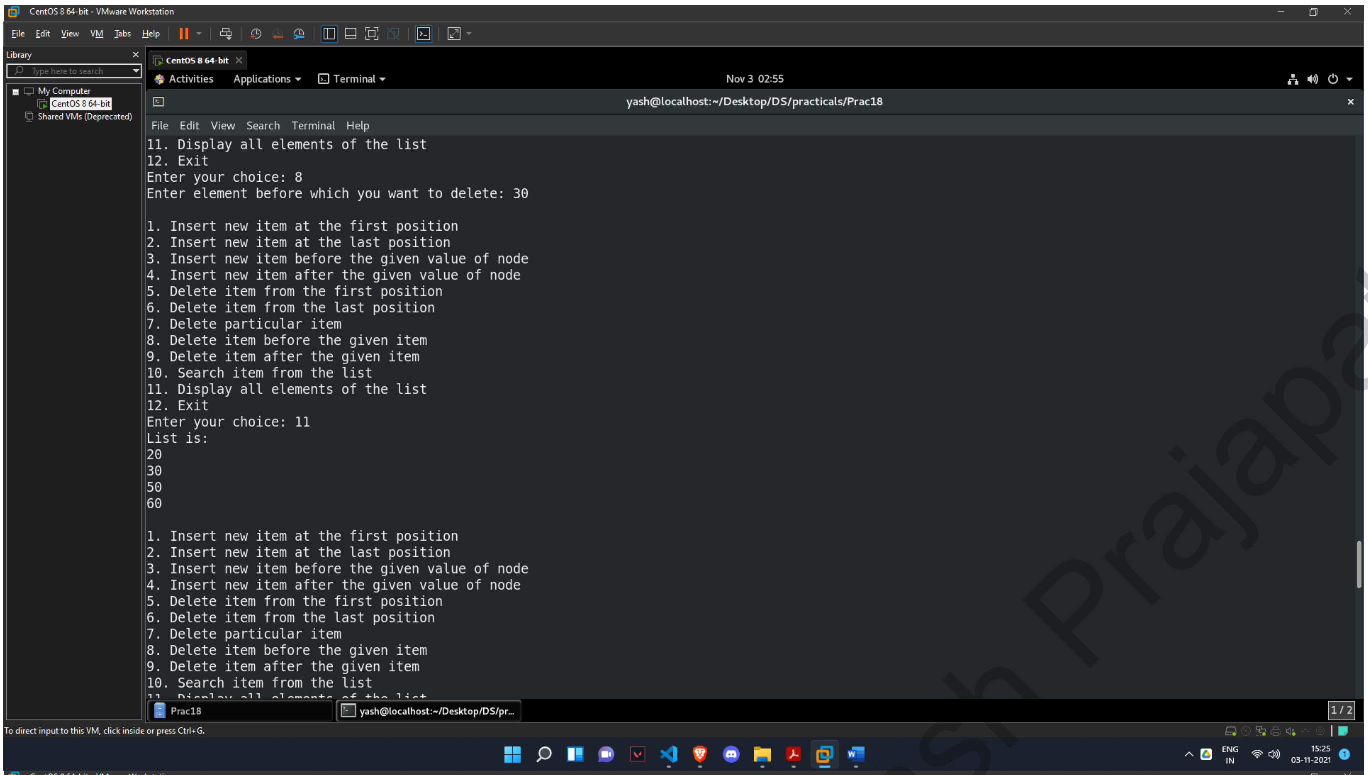
```
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 11
List is:
10
20
25
30
40
50
60
70

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 5

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
```



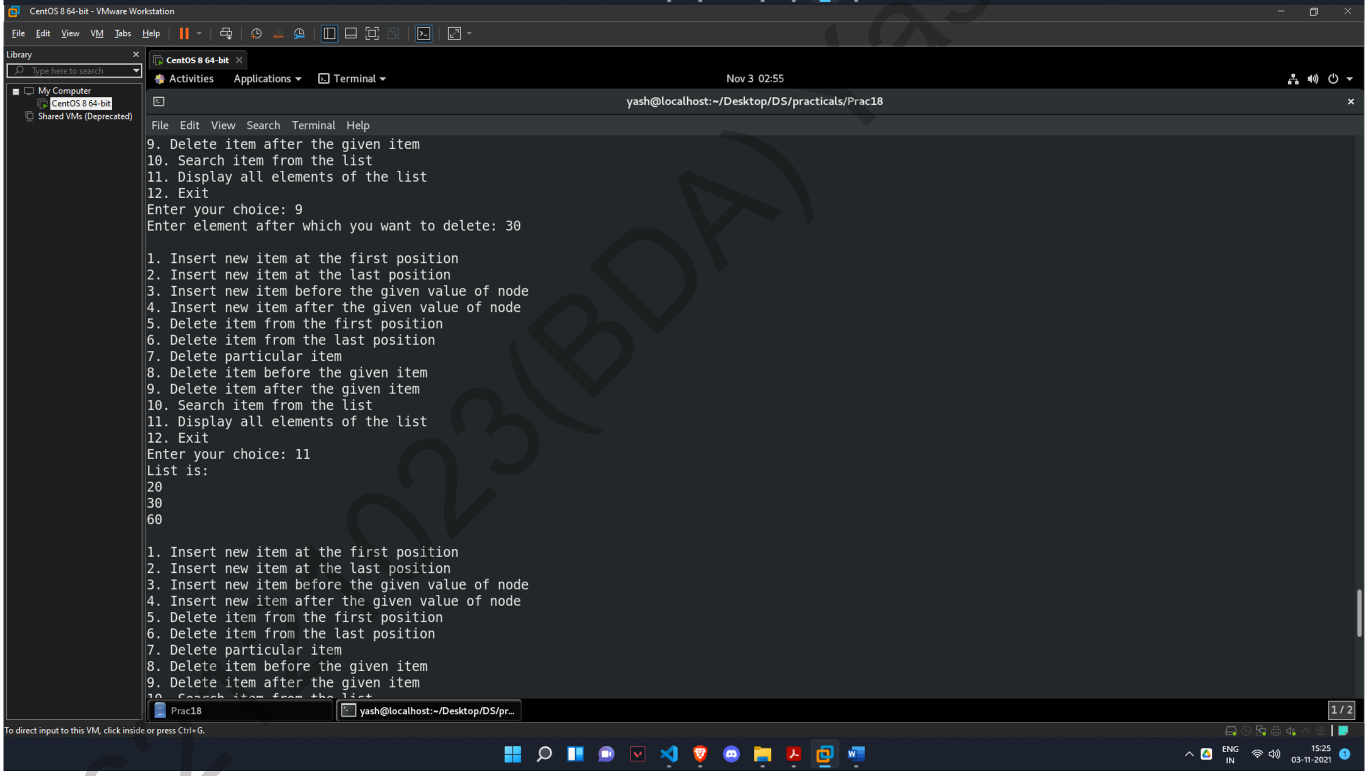
```
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 6

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 11
List is:
20
25
30
40
50
60

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
```



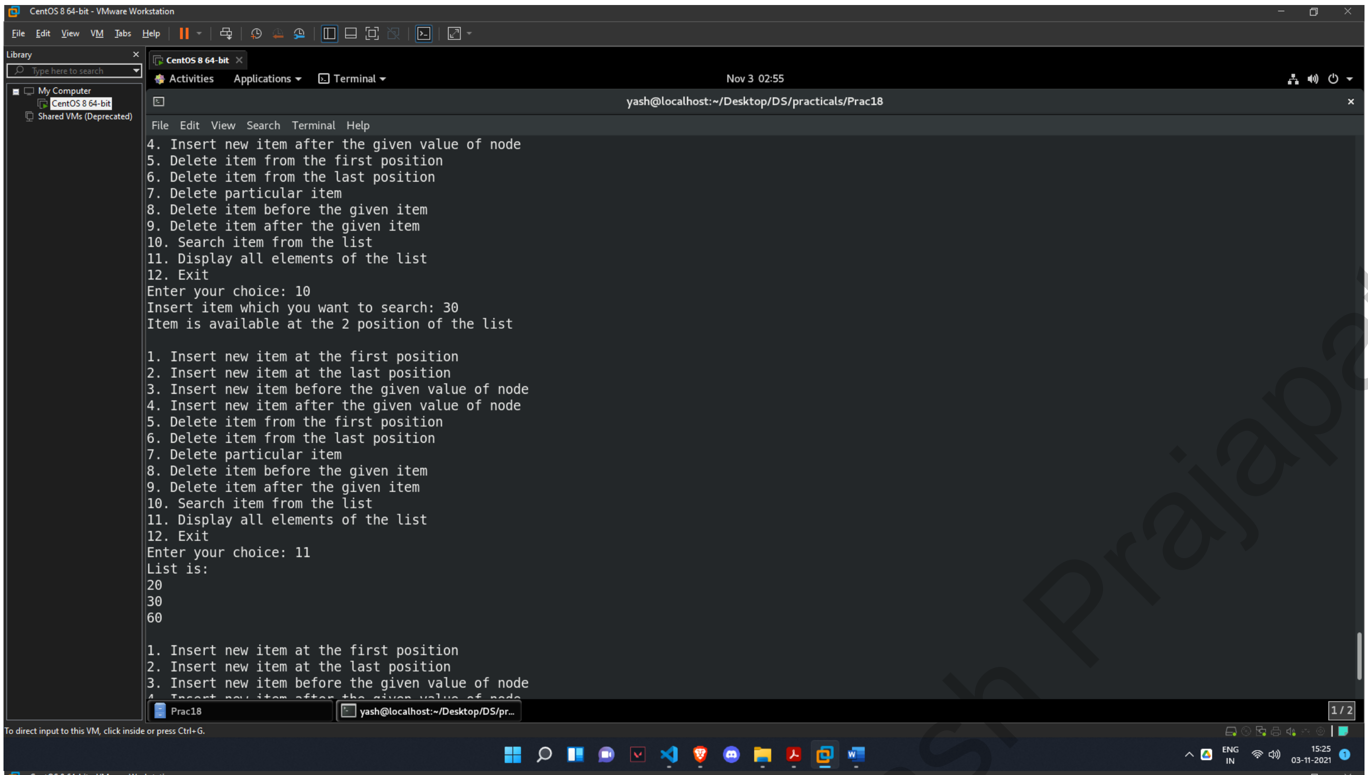
```
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 7
Enter element which you want to delete: 40

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 11
List is:
20
25
30
50
60

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
```



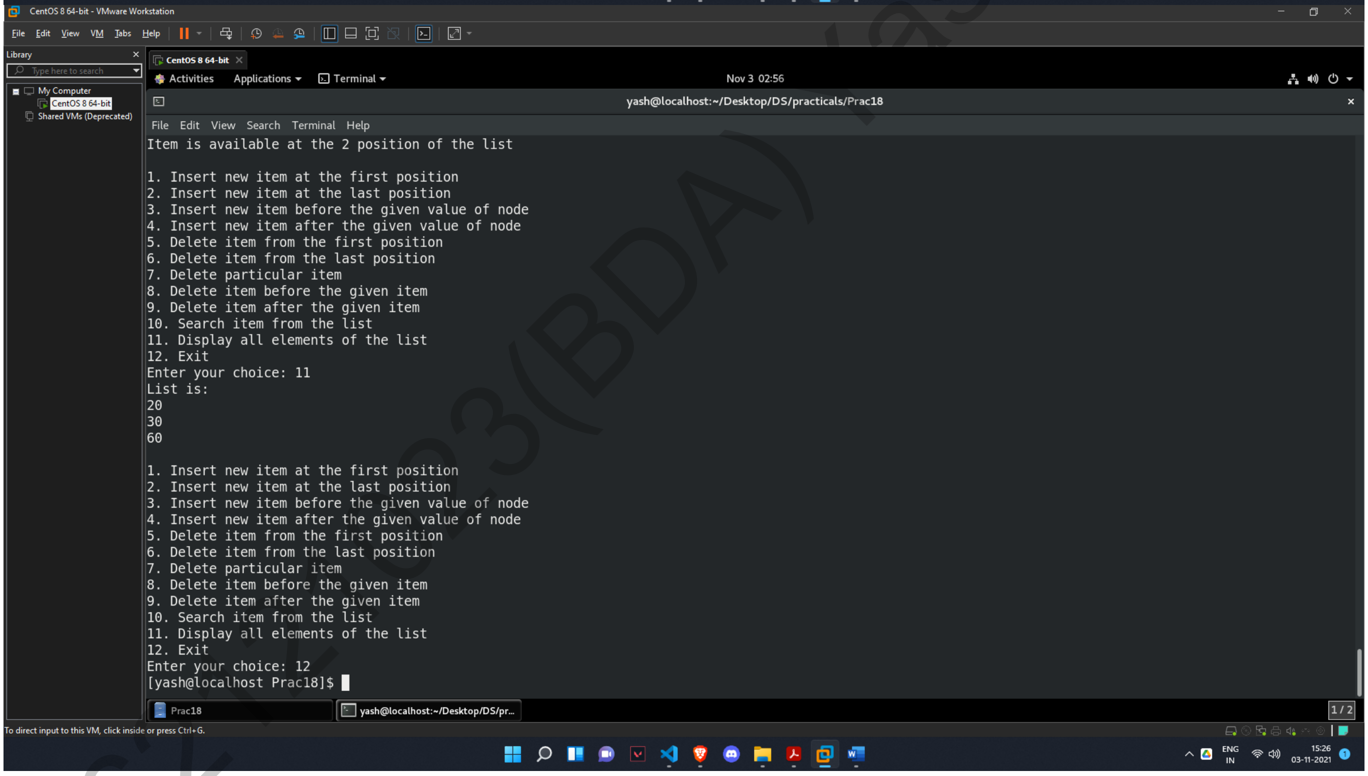
```
11. Display all elements of the list
12. Exit
Enter your choice: 8
Enter element before which you want to delete: 30

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 11
List is:
20
30
50
60

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
```



```
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 9
Enter element after which you want to delete: 30

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 11
List is:
20
30
60

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
```


CentOS 8 64-bit - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
CentOS 8 64-bit
Shared VMs (Deprecated)
Activities Applications Terminal
Nov 3 02:55
yash@localhost:~/Desktop/DS/practicals/Prac18
Prac18
1 / 2
To direct input to this VM, click inside or press Ctrl+G.

```
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 10
Insert item which you want to search: 30
Item is available at the 2 position of the list

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 11
List is:
20
30
60

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
```


CentOS 8 64-bit - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
CentOS 8 64-bit
Shared VMs (Deprecated)
Activities Applications Terminal
Nov 3 02:56
yash@localhost:~/Desktop/DS/practicals/Prac18
Prac18
1 / 2
To direct input to this VM, click inside or press Ctrl+G.

```
Item is available at the 2 position of the list

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 11
List is:
20
30
60

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 12
[yash@localhost Prac18]$ 
```

## Homework:

***Ques. Jay wants to add forward and backward button in his website which directs the addresses to previous and next from the current value. Refer the given scenario and provide appropriate solution using circular doubly linked list:***

**1. Insert new item at the first position**
**2. Insert new item at the last position**
**3. Insert new item before the given value of node**
**4. Insert new item after the given value of node**
**5. Delete item from the first position**
**6. Delete item from the last position**
**7. Delete particular item**
**8. Delete item before the given item**
**9. Delete item after the given item**
**10. Search item from the list**
**11. Display all elements of the list**

***SOLUTION***

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
        int data;
        struct node *previous, *next;
};
struct node *start;
void insert_begin()
{
        int yash;
        printf("Enter element you want to insert at first position: ");
        scanf("%d", &yash);
        struct node *new_node, *ptr;
        new_node = (struct node *)malloc(sizeof(struct node));
        new_node->data = yash;
        ptr = start;
        if (start == NULL)
        {
                start = new_node;
                new_node->next = start;
                new_node->previous = start;
        }
        else
```

```
		{
			start->previous = new_node;
			new_node->next = start;
			while (ptr->next != start)
			{
				ptr = ptr->next;
			}
			new_node->previous = ptr;
			start = new_node;
		}
	}
	void insert_end()
	{
		int yash;
		printf("Enter element you want to insert at last position: ");
		scanf("%d", &yash);
		struct node *new_node, *ptr;
		new_node = (struct node *)malloc(sizeof(struct node));
		ptr = start;
		new_node->data = yash;
		new_node->next = start;
		if (start == NULL)
		{
			start = new_node;
			new_node->previous = start;
		}
		else
		{
			while (ptr->next != start)
			{
				ptr = ptr->next;
			}
			new_node->previous = ptr;
			ptr->next = new_node;
		}
	}
	void insert_before()
	{
		int yash, item;
		printf("Enter element you want to insert: ");
		scanf("%d", &yash);
		printf("Enter element before which yo want to insert: ");
		scanf("%d", &item);
		struct node *new_node, *ptr, *before;
		new_node = (struct node *)malloc(sizeof(struct node));
		new_node->data = yash;
		ptr = start;
		before = ptr;
		while (ptr->data != item)
		{
			before = ptr;
			ptr = ptr->next;
		}
		before->next = new_node;
		new_node->previous = before;
		new_node->next = ptr;
		ptr->previous = new_node;
	}
	void insert_after()
	{
		int yash, item;
		printf("Enter element you want to insert: ");
		scanf("%d", &yash);
		printf("Enter element after which yo want to insert: ");
		scanf("%d", &item);
		struct node *new_node, *ptr;
		new_node = (struct node *)malloc(sizeof(struct node));
		new_node->data = yash;
		ptr = start;
		while (ptr->data != item)
		{
			ptr = ptr->next;
		}
		new_node->next = ptr->next;
		new_node->previous = ptr;
		ptr->next = new_node;
	}
	void delete_begin()
	{
		struct node *ptr;
		ptr = start;
		while (ptr->next != start)
		{
			ptr = ptr->next;
		}
		ptr->next = start->next;
		free(start);
		start = ptr->next;
```

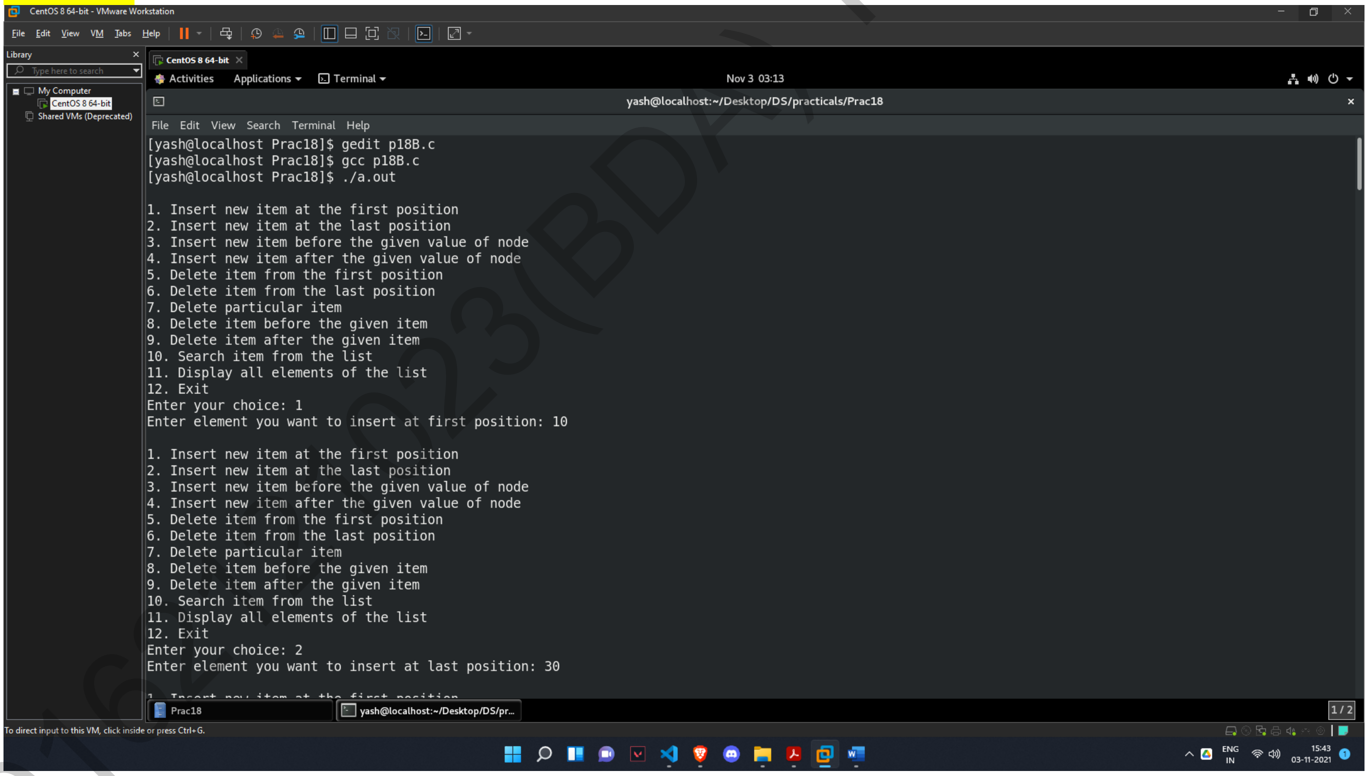
```
        ptr->next->previous = ptr;
}
void delete_last()
{
        struct node *ptr;
        ptr = start;
        while (ptr->next != start)
        {
                ptr = ptr->next;
        }
        ptr->previous->next = start;
        free(ptr);
}
void delete ()
{
        int yash;
        printf("Enter element which you want to delete: ");
        scanf("%d", &yash);
        struct node *ptr;
        ptr = start;
        while (ptr->data != yash)
        {
                ptr = ptr->next;
        }
        ptr->previous->next = ptr->next;
        ptr->next->previous = ptr->previous;
        free(ptr);
}
void delete_before()
{
        int yash;
        printf("Enter element before which you want to delete: ");
        scanf("%d", &yash);
        struct node *ptr, *before;
        ptr = start;
        before = ptr;
        while (ptr->data != yash)
        {
                before = ptr;
                ptr = ptr->next;
        }
        ptr->previous = ptr->previous->previous;
        before->previous->next = ptr;
        free(before);
```

```
}
void delete_after()
{
        int yash;
        printf("Enter element after which you want to delete: ");
        scanf("%d", &yash);
        struct node *ptr, *after;
        ptr = start;
        while (ptr->data != yash)
        {
                ptr = ptr->next;
        }
        after = ptr->next;
        ptr->next = after->next;
        after->next->previous = after->previous;
        free(after);
}
void search()
{
        int yash, pos = 0;
        printf("Insert item which you want to search: ");
        scanf("%d", &yash);
        struct node *ptr;
        ptr = start;
        while (ptr->next != start)
        {
                if (ptr->data == yash)
                {
                        break;
                }
                else
                {
                        ptr = ptr->next;
                        pos++;
                }
        }
        printf("Item is available at the %d position of the list\n", pos + 1);
}
void display()
{
        printf("List is:\n");
        struct node *ptr;
        ptr = start;
        while (ptr->next != start)
```

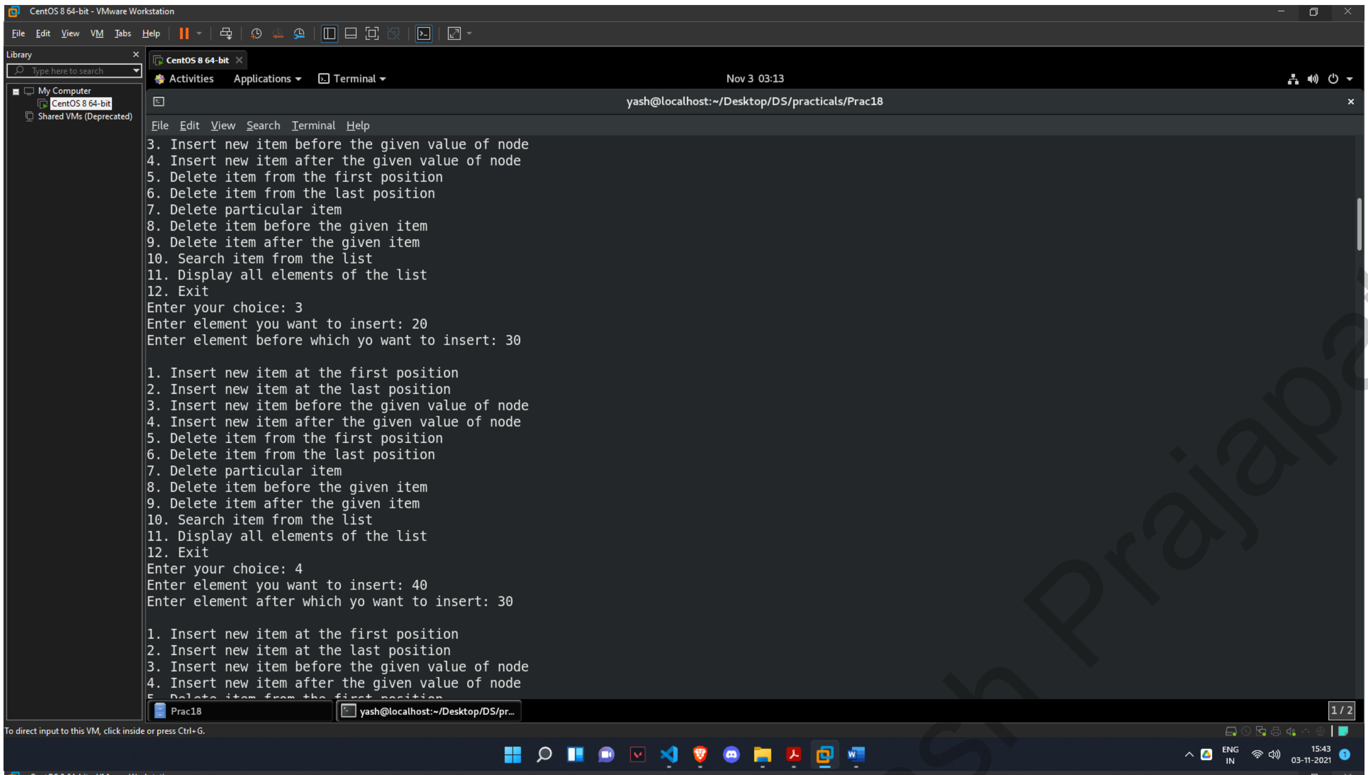
```
		{
			printf("%d\t", ptr->data);
			ptr = ptr->next;
		}
		printf("%d\t", ptr->data);
	}
	int main()
	{
		int choice;
		for (;;)
		{
			printf("\n1. Insert new item at the first position\n2. Insert new item at the
	last position\n3. Insert new item before the given value of node\n4. Insert new item after
	the given value of node\n5. Delete item from the first position\n6. Delete item from the
	last position\n7. Delete particular item\n8. Delete item before the given item\n9. Delete
	item after the given item\n10. Search item from the list\n11. Display all elements of the
	list\n12. Exit\n");
			printf("Enter your choice: ");
			scanf("%d", &choice);
			switch (choice)
			{
			case 1:
				insert_begin();
				break;
			case 2:
				insert_end();
				break;
			case 3:
				insert_before();
				break;
			case 4:
				insert_after();
				break;
			case 5:
				delete_begin();
				break;
			case 6:
				delete_last();
				break;
			case 7:
				delete ();
				break;
			case 8:
				delete_before();
```

```
                    break;
            case 9:
                    delete_after();
                    break;
            case 10:
                    search();
                    break;
            case 11:
                    display();
                    break;
            case 12:
                    exit(0);
                    break;
            }
    }
    return 0;
}
```

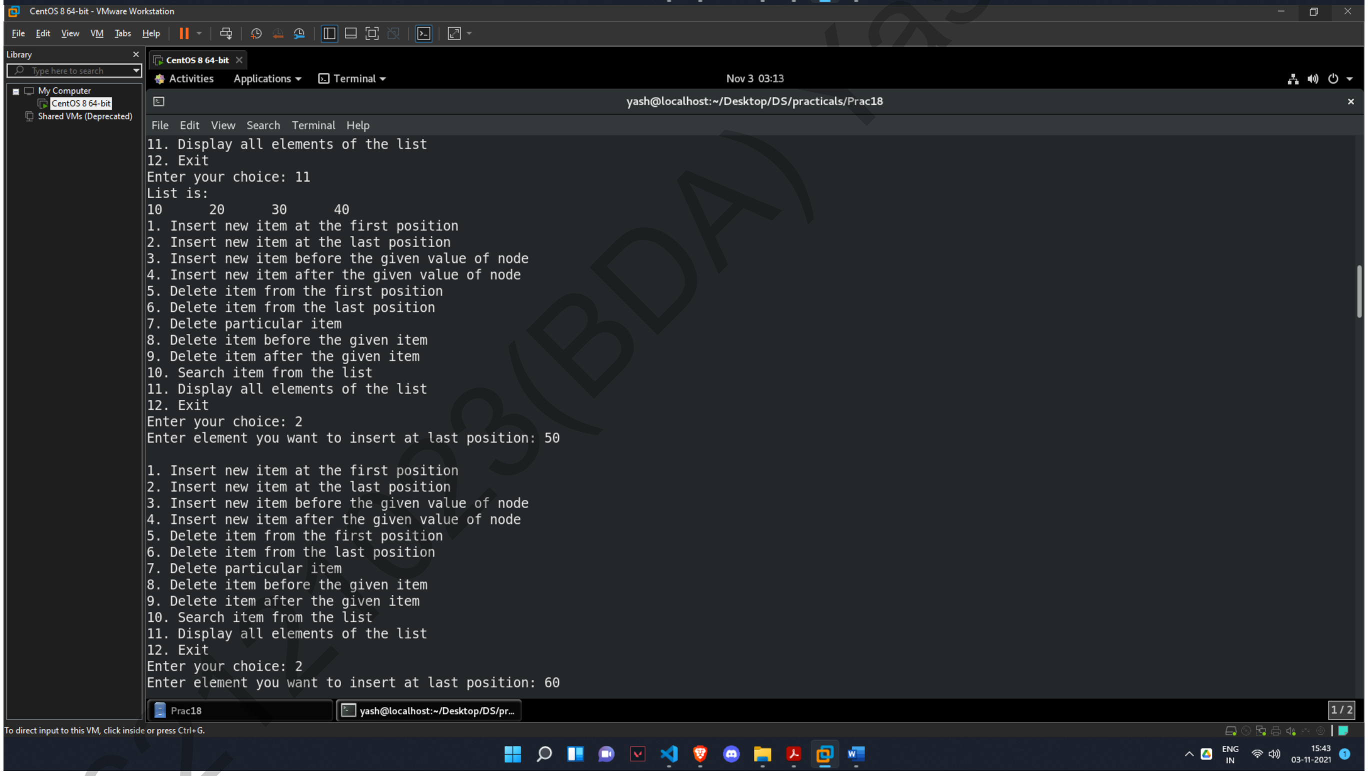
***OUTPUT***


CentOS 8 64-bit - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
CentOS 8 64-bit
Shared VMs (Deprecated)
CentOS 8 64-bit
Activities
Applications
Terminal
Nov 3 03:13
yash@localhost:~/Desktop/DS/practicals/Prac18
File Edit View Search Terminal Help
Prac18
yash@localhost:~/Desktop/DS/pr...
1 / 2
To direct input to this VM, click inside or press Ctrl+G.
ENG IN
15:43
03-11-2021

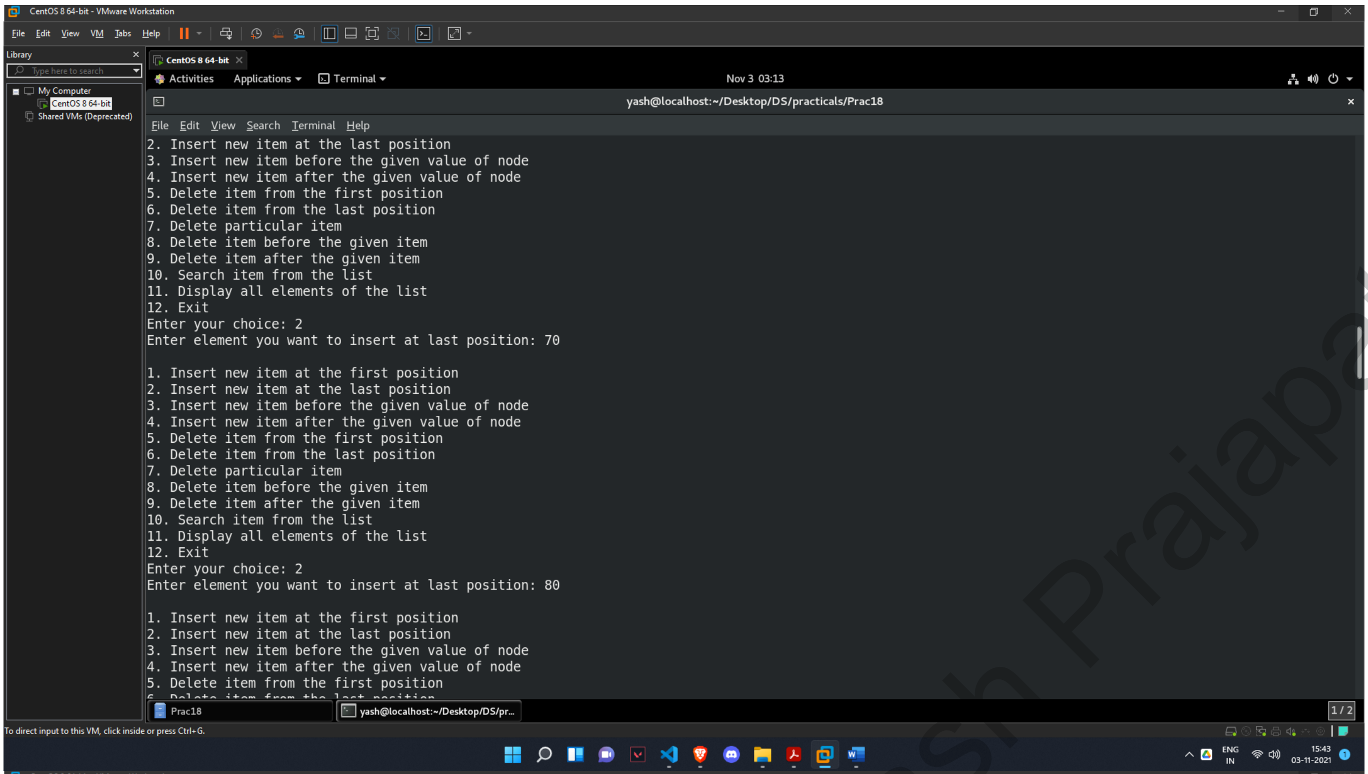
```
[yash@localhost Prac18]$ gedit p18B.c
[yash@localhost Prac18]$ gcc p18B.c
[yash@localhost Prac18]$ ./a.out

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 1
Enter element you want to insert at first position: 10

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 2
Enter element you want to insert at last position: 30

1. Insert new item at the first position
```



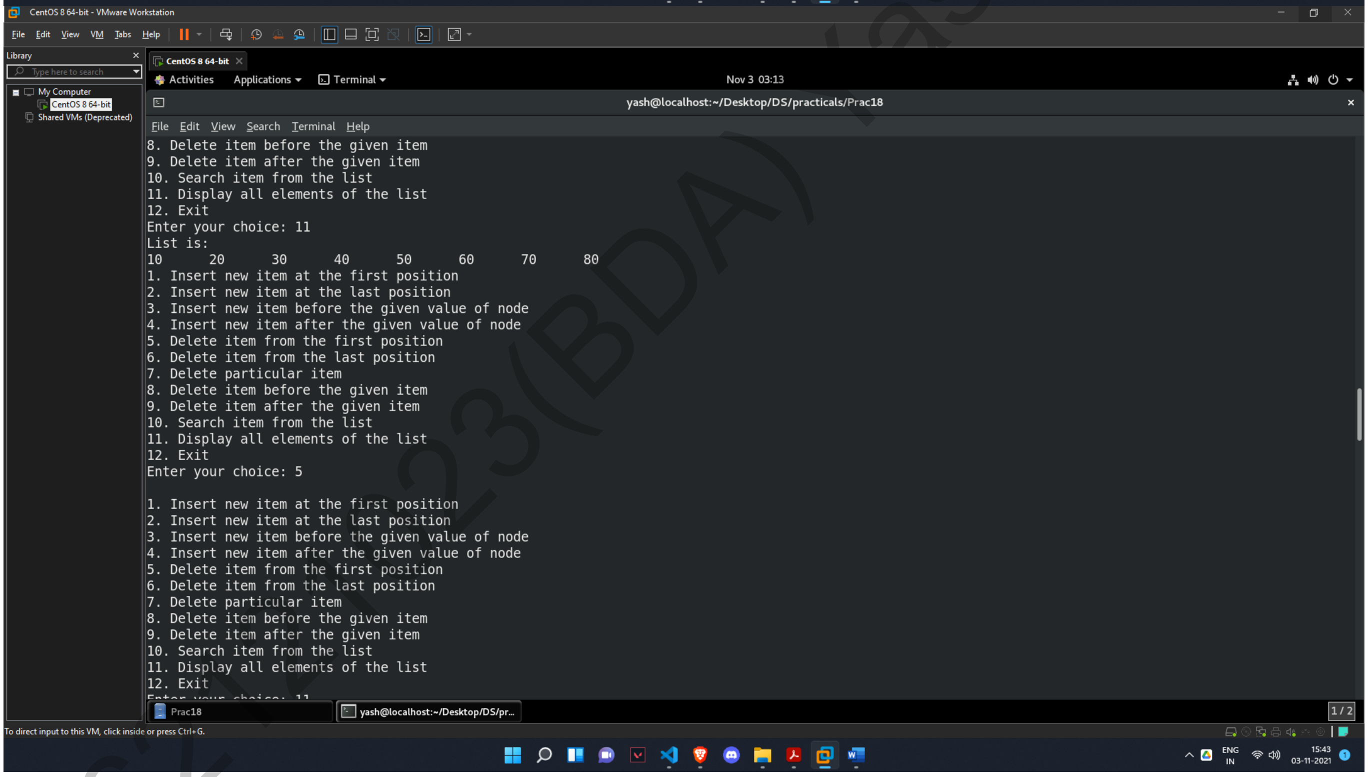
```
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 3
Enter element you want to insert: 20
Enter element before which yo want to insert: 30

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 4
Enter element you want to insert: 40
Enter element after which yo want to insert: 30

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
```



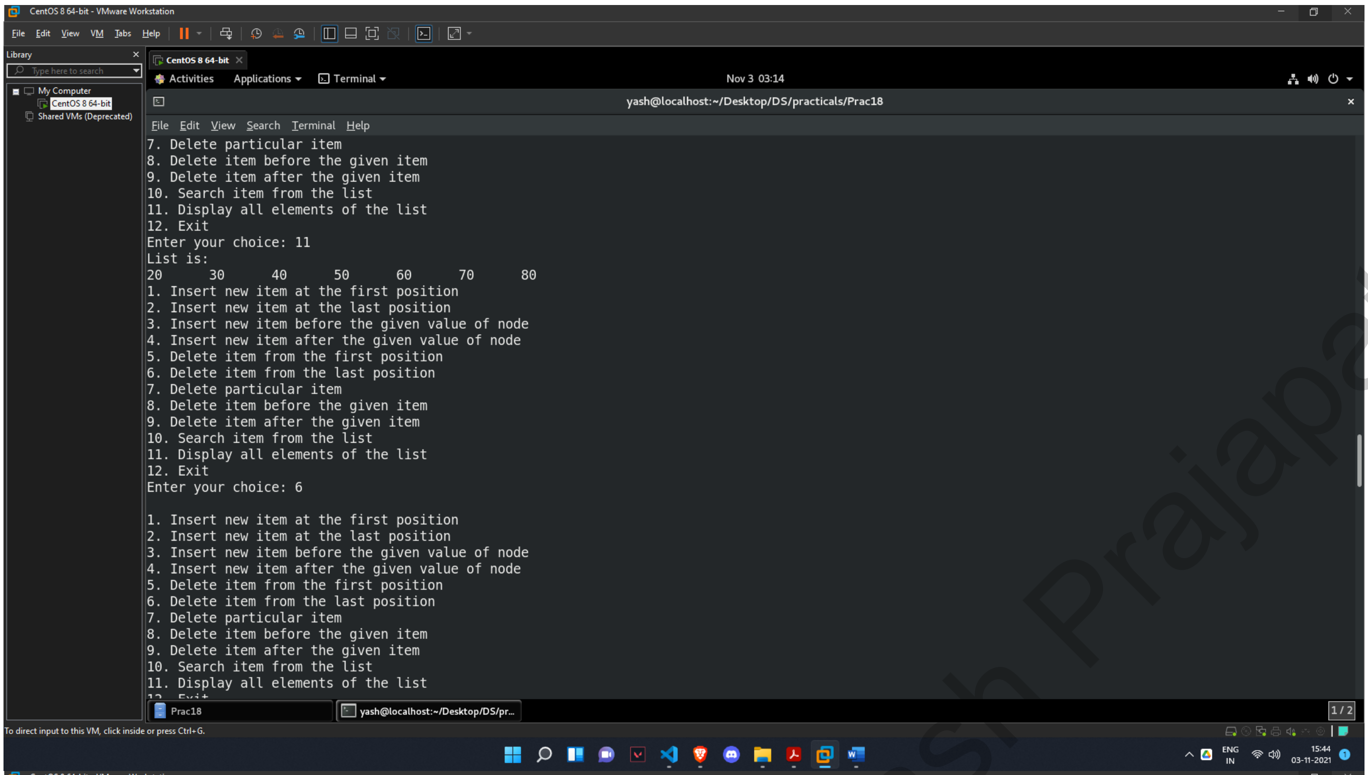
```
11. Display all elements of the list
12. Exit
Enter your choice: 11
List is:
10      20      30      40
1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 2
Enter element you want to insert at last position: 50

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 2
Enter element you want to insert at last position: 60
```



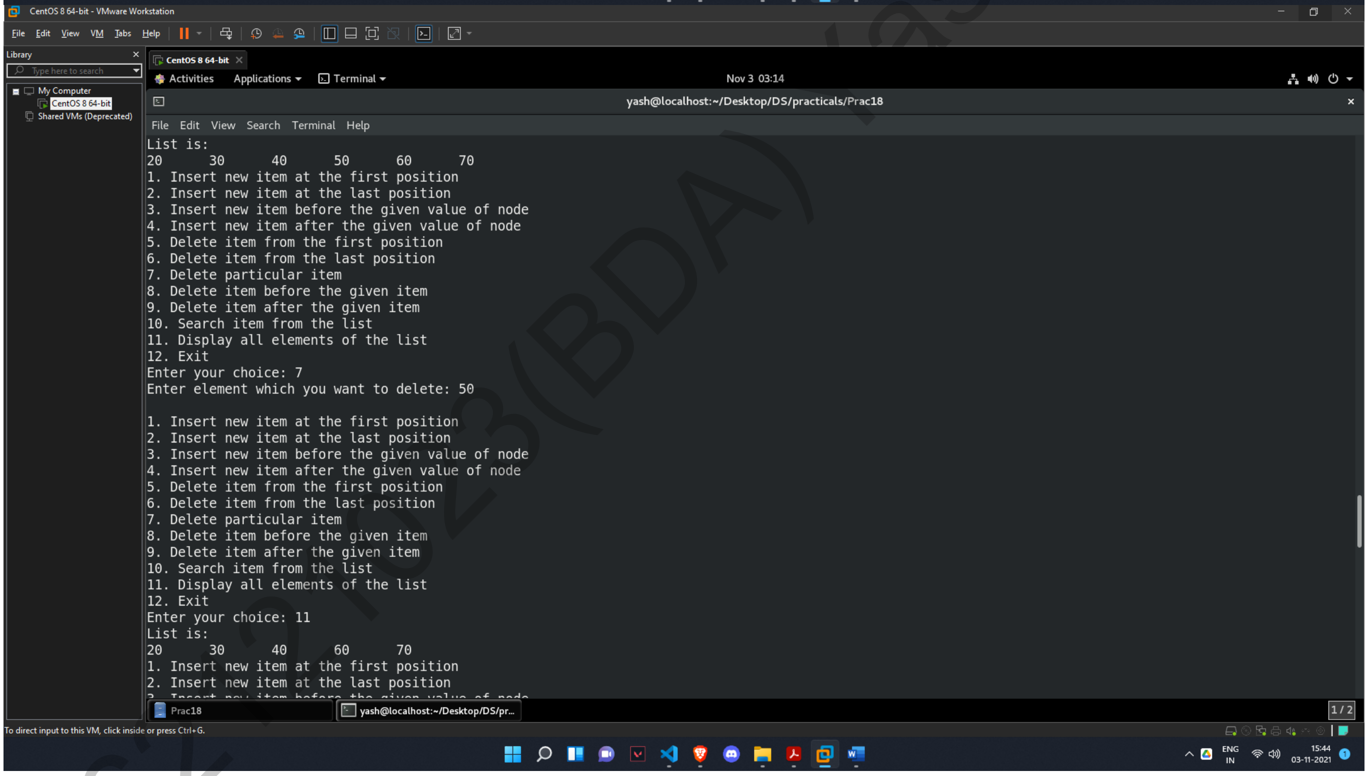
```
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 2
Enter element you want to insert at last position: 70

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 2
Enter element you want to insert at last position: 80

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
```



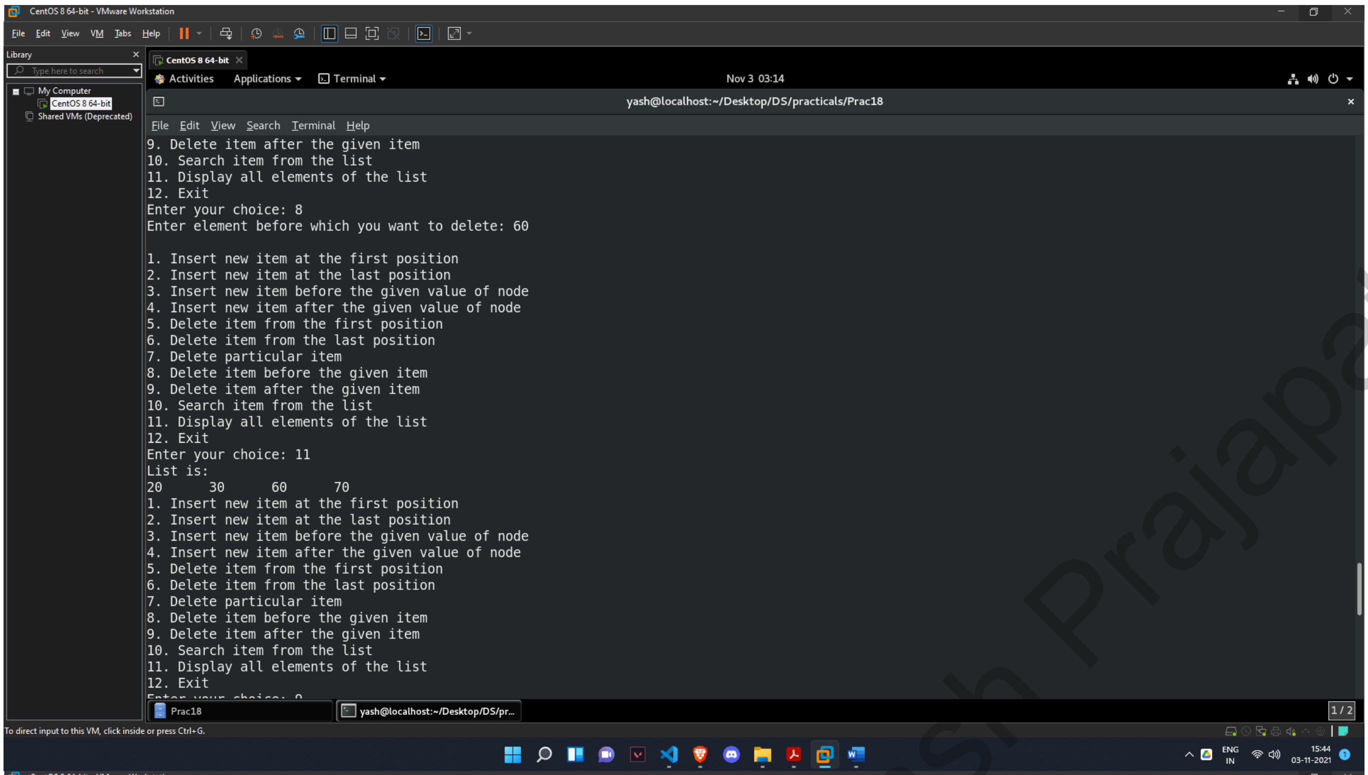
```
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 11
List is:
10      20      30      40      50      60      70      80
1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 5

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
```



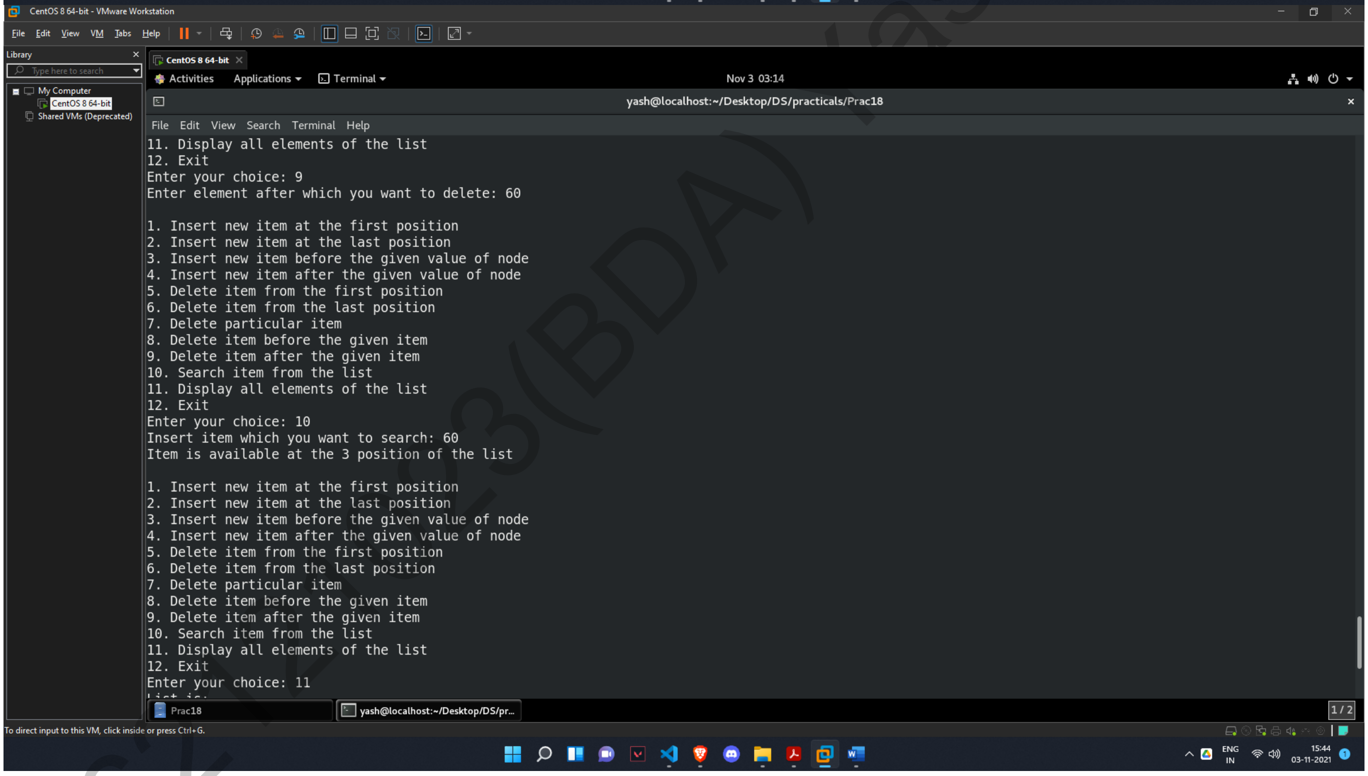
```
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 11
List is:
20      30      40      50      60      70      80
1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 6

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
```



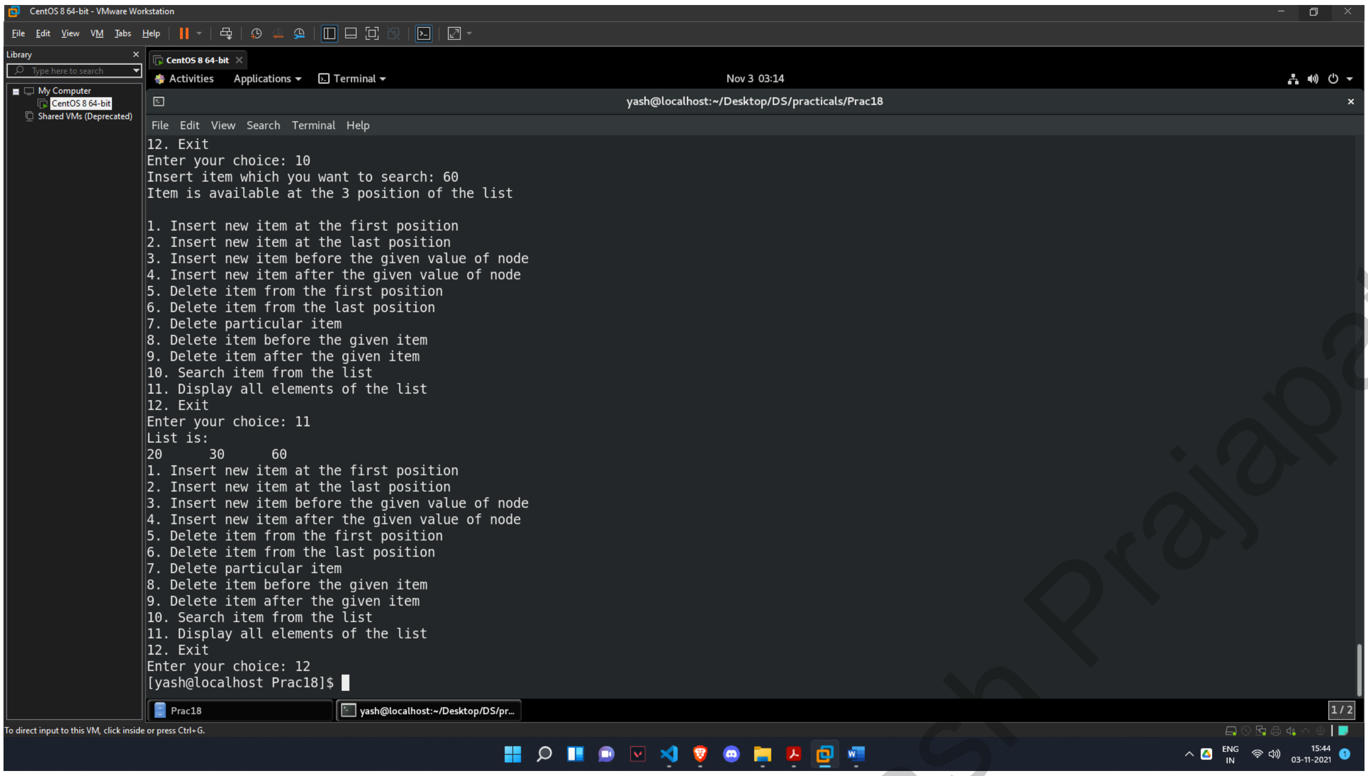
```
List is:
20      30      40      50      60      70
1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 7
Enter element which you want to delete: 50

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 11
List is:
20      30      40      60      70
1. Insert new item at the first position
2. Insert new item at the last position
```


CentOS 8 64-bit - VMware Workstation
Activities
Applications
Terminal
Nov 3 03:14
yash@localhost:~/Desktop/DS/practicals/Prac18

```
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 8
Enter element before which you want to delete: 60

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 11
List is:
20      30      60      70
1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 9
```


CentOS 8 64-bit - VMware Workstation
Activities
Applications
Terminal
Nov 3 03:14
yash@localhost:~/Desktop/DS/practicals/Prac18

```
11. Display all elements of the list
12. Exit
Enter your choice: 9
Enter element after which you want to delete: 60

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 10
Insert item which you want to search: 60
Item is available at the 3 position of the list

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 11
List is:
```



```
12. Exit
Enter your choice: 10
Insert item which you want to search: 60
Item is available at the 3 position of the list

1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 11
List is:
20      30      60
1. Insert new item at the first position
2. Insert new item at the last position
3. Insert new item before the given value of node
4. Insert new item after the given value of node
5. Delete item from the first position
6. Delete item from the last position
7. Delete particular item
8. Delete item before the given item
9. Delete item after the given item
10. Search item from the list
11. Display all elements of the list
12. Exit
Enter your choice: 12
[yash@localhost Prac18]$
```