

```
C:\Users\Devsuman\Desktop X + ×
1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
10

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
20

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
2
item deleted =10

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
3
Contents of queue
20

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
4

Process returned 0 (0x0) execution time : 22.206 s
Press any key to continue.
|
```



```
C:\Users\Devsuman\Desktop X + - 0 X
```

\*\*\*\*MAIN MENU\*\*\*\*  
1.Insert an element  
2.Delete an element  
3.Peek  
4.Display the queue  
5.Exit  
Enter your option:1  
Enter the number to be inserted in the queue:18

\*\*\*\*MAIN MENU\*\*\*\*  
1.Insert an element  
2.Delete an element  
3.Peek  
4.Display the queue  
5.Exit  
Enter your option:1  
Enter the number to be inserted in the queue:28

\*\*\*\*MAIN MENU\*\*\*\*  
1.Insert an element  
2.Delete an element  
3.Peek  
4.Display the queue  
5.Exit  
Enter your option:1  
Enter the number to be inserted in the queue:38

\*\*\*\*MAIN MENU\*\*\*\*  
1.Insert an element  
2.Delete an element  
3.Peek  
4.Display the queue  
5.Exit  
Enter your option:1  
Enter the number to be inserted in the queue:25

OVERFLOW

\*\*\*\*MAIN MENU\*\*\*\*  
1.Insert an element  
2.Delete an element  
3.Peek  
4.Display the queue  
5.Exit  
Enter your option:2  
The number deleted is ::18

\*\*\*\*MAIN MENU\*\*\*\*  
1.Insert an element  
2.Delete an element  
3.Peek  
4.Display the queue  
5.Exit  
Enter your option:3  
The first value in queue is:28

\*\*\*\*MAIN MENU\*\*\*\*  
1.Insert an element  
2.Delete an element  
3.Peek  
4.Display the queue  
5.Exit

24°C Clear

Search

22:17 23-01-2024

```
34
-----
34
75
56
87
-----
value 89 added successful at 3
34
75
89
56
87
Process returned 0 (0x0) execution time : 0.060 s
Press any key to continue.
```



Search



ENG  
IN



22:56  
23-01-2024

```
C:\Users\Devsuman\Desktop > + ×  
Enter operation:  
1.create  
2.display  
3.delete at beginning  
4.delete at end  
5.delete at position  
6.-1 to end  
1  
enter the number of elements:  
5  
Enter the element 1:  
10  
Enter the element 2:  
20  
Enter the element 3:  
30  
Enter the element 4:  
40  
Enter the element 5:  
50  
Enter operation:  
1.create  
2.display  
3.delete at beginnnning  
4.delete at end  
5.delete at position  
6.-1 to end  
2  
The elements are:  
10  
20  
30  
40  
50  
Enter operation:  
1.create  
2.display  
3.delete at beginning  
4.delete at end  
5.delete at position  
6.-1 to end  
3  
Enter operation:  
1.create  
2.display  
3.delete at beginnnning  
4.delete at end  
5.delete at position  
6.-1 to end  
4  
Enter operation:  
1.create  
2.display  
3.delete at beginning  
4.delete at end  
5.delete at position  
6.-1 to end  
5  
enter the position:  
3  
3  
Enter operation:  
1.create  
2.display  
3.delete at beginning  
4.delete at end  
5.delete at position  
6.-1 to end  
-1  
operation completed!  
  
Process returned 0 (0x0) execution time : 37.571 s  
Press any key to continue.  
  
1 Upcoming Earnings   
ENG IN 23:01:2024
```

# DS LAB-4

22.1.24

- beg
- end
- At Pos
- del at end, At beg, at pos

# include <stdio.h>

# include <stdlib.h>

struct node

{

int data;

struct node \*next;

};

Void PrintData (Struct node \*head)

{

if (head == null) {

Dprintf ("The list is empty");

}

else

{

Struct node \*ptr = head;

While (ptr != NULL)

{

printf ("%d\n", ptr -> data);

ptr = ptr -> next;

}

{

{

Void insertbeg (Struct node \*\* head, int value)

{

Struct node \* temp = (Struct node \*)  
malloc ( sizeof ( Struct node ));  
temp → data = value;  
temp → next = \* head;  
\* head = temp;

Void insertend (Struct node \*\* head,  
int value)

{

Struct node \* p1 = head;  
Struct node \* temp = (Struct node \*)  
malloc ( sizeof ( Struct node ));  
temp → data = value;  
temp → next = NULL;  
while ( p1 → next != NULL ) {  
p1 = p1 → next;  
}

p1 → next = temp;

Void insertatpos (Struct node \* head,  
int value, int pos)

{

Struct node \* Ptg \* Ptg2;

Struct node \* temp = (Struct node \*);

malloc (size of (Struct node));

temp → data = Value;

temp → next = NULL;

int position; pos;

Ptg = head;

While (Pos != 1)

Ptg2 = Ptg;

Ptg = Ptg → next;

Pos = -;

temp → next = Ptg2 → next;

Ptg2 → next = temp;

printf ("Value %d added successful  
at %d \n", value, position);

Void deleq (Struct node \*\* head)

Struct node \* Ptg;

If (\*head == NULL)

printf ("The list is empty");

```
else {  
    Ptr = * head;  
    * head = (* head) → next;  
    free(Ptr);  
    Ptr = NVLL;  
}
```

```
void delend (Struct node * head)  
{
```

```
Struct node * Ptr, * Ptr2;  
if (head == NVLL)
```

```
printf ("The list is empty");
```

```
}
```

```
else
```

```
Ptr = head;
```

```
while (Ptr → next != NVLL)
```

```
    Ptr2 = Ptr;
```

```
    Ptr = Ptr → next;
```

```
}
```

```
    Ptr2 → next = NULL;
```

```
    free(Ptr);
```

```
}
```

```
}
```

```
Void delatpos (Struct node * head, int pos)
```

Start node \*P<sub>tr</sub>\* P<sub>tr2</sub>  
if (head == NULL)  
    {  
        Printf ("The list is empty");  
    }  
    else if (Pos == 1)  
    {  
        P<sub>tr</sub> = head;  
        free(P<sub>tr</sub>);  
        P<sub>tr</sub> = NULL;

}  
else  
    P<sub>tr</sub> = head;  
    P<sub>tr2</sub> = head;  
    while (Pos != 1)  
    {  
        P<sub>tr2</sub> = P<sub>tr</sub>;  
        P<sub>tr</sub> = P<sub>tr</sub> → next;  
        Pos--;

}  
P<sub>tr2</sub> → next → P<sub>tr</sub> → next;  
free(P<sub>tr</sub>);  
P<sub>tr</sub> = NULL;

}  
} /\* else (Pos != 1) \*/

11

int main ()  
{  
 struct node \*head = null;  
 insert at beg (& head , 34);  
 Print Data (head);  
 printf (" - - - - - \n");  
 insert end (head , 75);  
 insert End (head , 56);  
 insert and (head , 87);  
 PrintData (head);  
 printf (" - - - - - \n");  
 insert At Pos (head , 89 , 3);  
 Print data (head);  
 printf (" - - - - - \n");  
 deleteq (& head);  
 printf (" - - - - - \n");  
 deleteq (& head);  
 PrintData (head);  
 printf (" - - - - - \n");  
 delat pos (head , 2);  
 Print data (head);  
 printf (" - - - - - \n");  
}

Output

Value 34 added successfully at the beginning

34

-----

Value 75 added successfully at the end

value

output

Value 56 added successfully at the end.

Value 87 is added successfully at the end

34

75

56

87

Value 89 added successfully at the end

34

75

56

87

— / / —

75

89

56

87

75

89

56

75

87