OUTPUT

Student Roll

(A) Insert Front
(B) Delete Front
(C) Insert End
(D) Delete End
(E) Add After Roll
(F) Delete Roll
(G) Show List
(H) EXIT

Option: A
Enter name: Anil
Enter roll: 10
Enter marks: 44

Option: C
Enter name: Binu
Enter roll: 20
Enter marks: 55

Option: G

Roll	Name	Marks
10	Anil	44.00
20	Binu	55.00

Option: E
Enter roll: 10
Enter name: Cathy
Enter roll: 30
Enter marks: 66

Option: G

Roll	Name	Marks
10	Anil	44.00
30	Cathy	66.00
20	Binu	55.00

Option: F
Enter Roll: 20

Option: G

Roll	Name	Marks
10	Anil	44.00
30	Cathy	66.00

Q Create a singly linked list to store the name, roll no:, marks obtained by n students. 17-Jan-2022
Implement the following functions:
(i) Insert student at front of list
(ii) Insert student at end of list
(iii) Insert student after a particular student (roll no:)
(iv) Delete student at front of list
(v) Delete student at end of list
(vi) Delete student having particular roll number.

ALGORITHM

1. Declare a structure "Node" with name, roll, mark, next.

I. makeNode()
1. Create a new node, newNode
2. INPUT name, roll number, mark into newNode at name, roll, mark.
3. return newNode

II. addFront()
1. Get n from makeNode()
2. Set n→next to NULL if (HEAD==NULL), else to HEAD.
3. HEAD = n

III. removeFront()
1. if (HEAD == NULL): return.
2. temp = HEAD → next
3. Delete HEAD
4. HEAD = temp.

```
Option: B
Option: 67
  Roll        Name        Marks
  30          Cathy       66.00

Option: A
  Enter Name: Daniel
  Enter roll: 40
  Enter marks: 77

Option: D
Option: 67
  Roll        Name        Marks
  40          Daniel      77.00

Option: H
```

IV. addEnd()

1. Get n from makeNode()
2. if (HEAD == NULL):
   HEAD = n
3. else:
   end = HEAD
   while (end → next ≠ NULL):
      end = end → next
   end → next = n

V. removeEnd()

1. if (HEAD == NULL): return
2. end = HEAD
3. while (end → next ≠ NULL):
   secEnd = end
   end = end → next.
4. if (secEnd == NULL): HEAD = NULL
   else: secEnd → next = NULL
5. Delete end

VI. addAfterRoll(rk)

1. Get n from makeNode()
2. pos = HEAD
3. while (pos → roll ≠ rk AND pos ≠ NULL):
   pos = pos → next.
4. if (pos == NULL): return
5. n → next = pos → next
6. pos → next = n

VII. deleteRoll(rk)

1. pos = HEAD
2. while (pos→roll ≠ rk AND pos ≠ NULL):
   pos = pos→next
3. if (pos == NULL): return
4. if (secPos == NULL): HEAD = pos→next
   else: secPos→next = pos→next
5. Delete pos.

VIII. showList()

1. n = HEAD
2. while (n ≠ NULL):
   DISPLAY n→name, n→roll, n→mark
   n = n→next

IX. main()

1. Display a menu with options 1-6
2. INPUT roll number as rk if necessary
3. Call corresponding function from list:
   addFront(), removeFront(), addEnd(), removeEnd(), addAfterRoll(rk), deleteRoll(rk), showList(),

RESULT

Program is executed successfully and output is obtained.