The background of the slide is decorated with various light green 3D geometric shapes. These include a small cube in the top left, a large sphere on the left, a square frame in the upper center, a sphere in the upper right, a wavy line in the top right, a cone on the right, a cylinder in the bottom left, a large sphere in the bottom center, a cone on the bottom right, and a small cube in the bottom right. The shapes are scattered across the slide, creating a modern and abstract aesthetic.

Data Structure

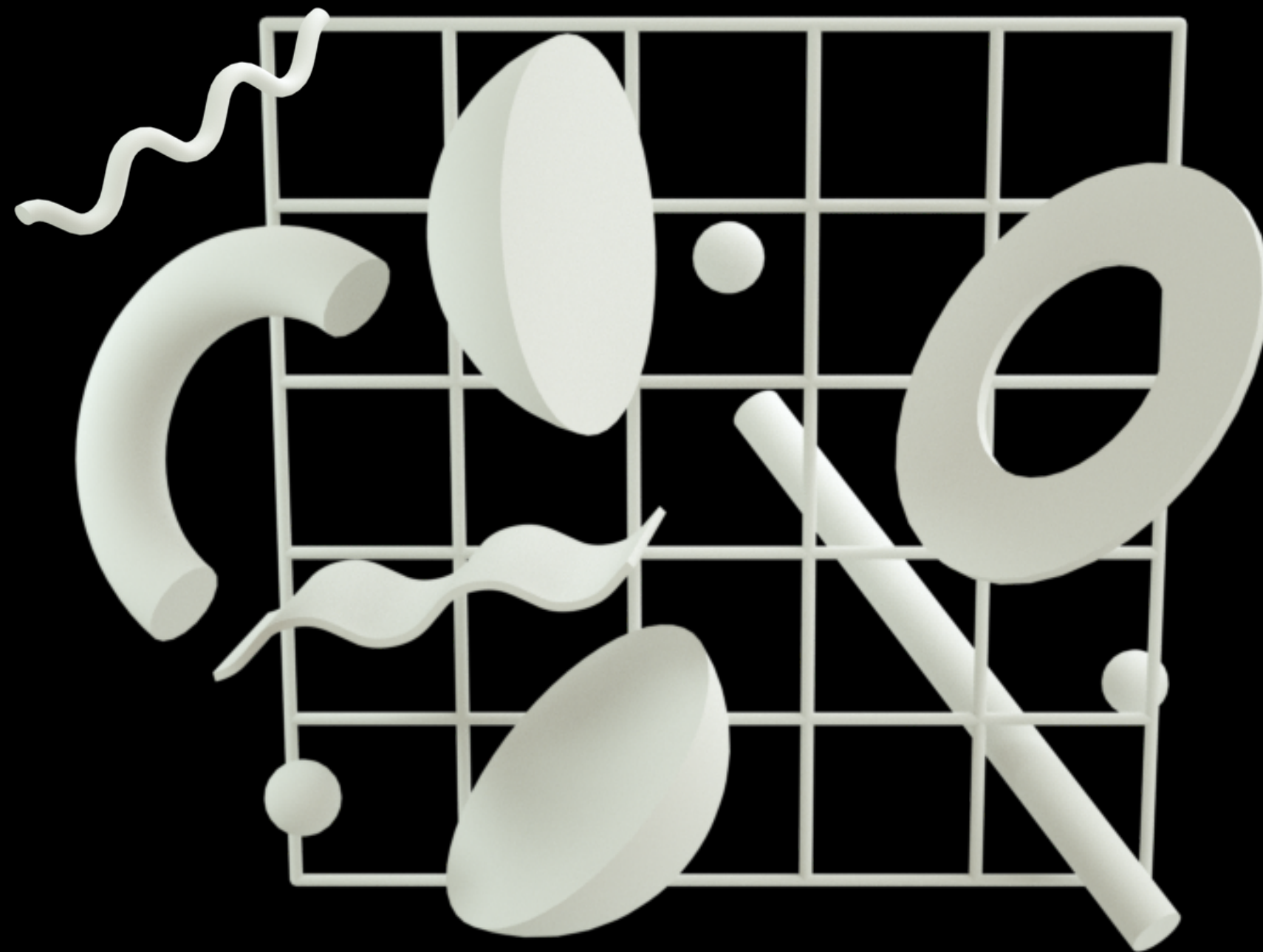
Afsana Begum
Assistant Professor,
Department of Software Engineering,
Daffodil International University,

Dynamite

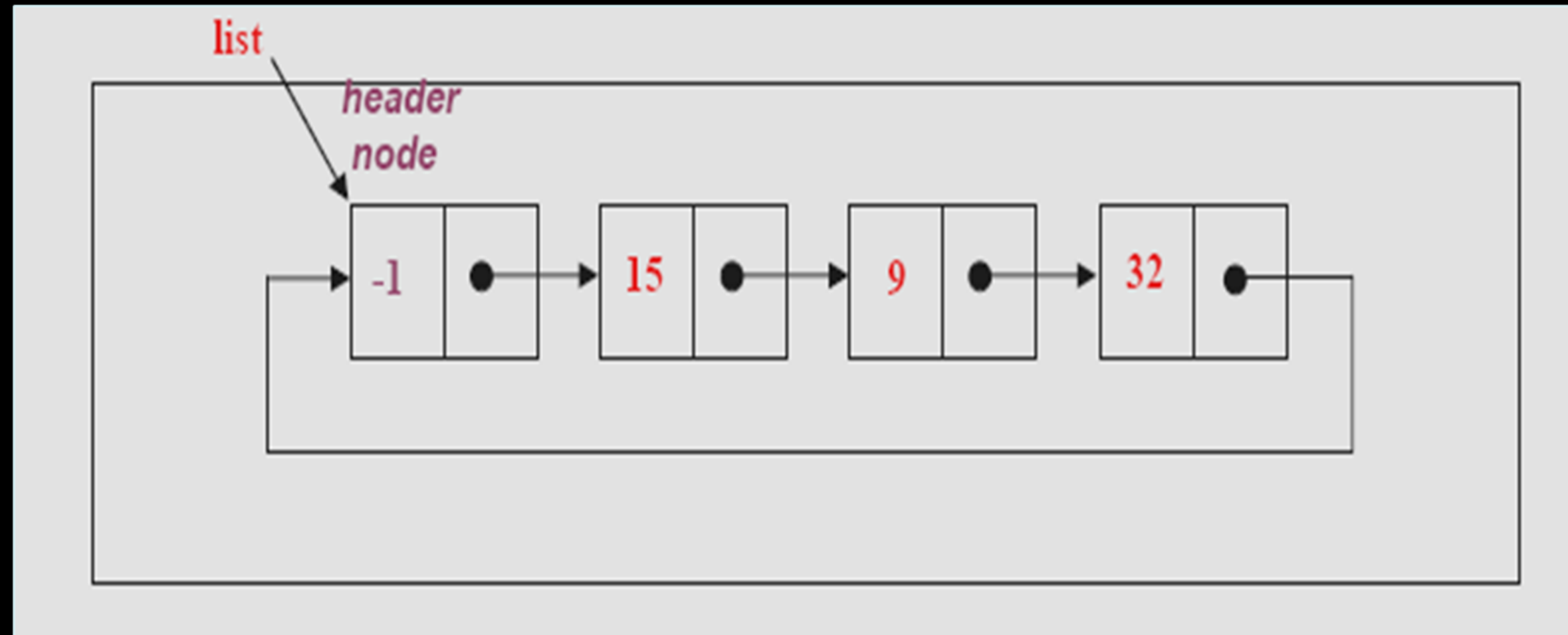
Name: Simanto Kumar Roy
Id: 221-35-909

Name: Minhazul Islam
Id: 221-35-932

Name: Sudip Datta
Id: 221-35-880

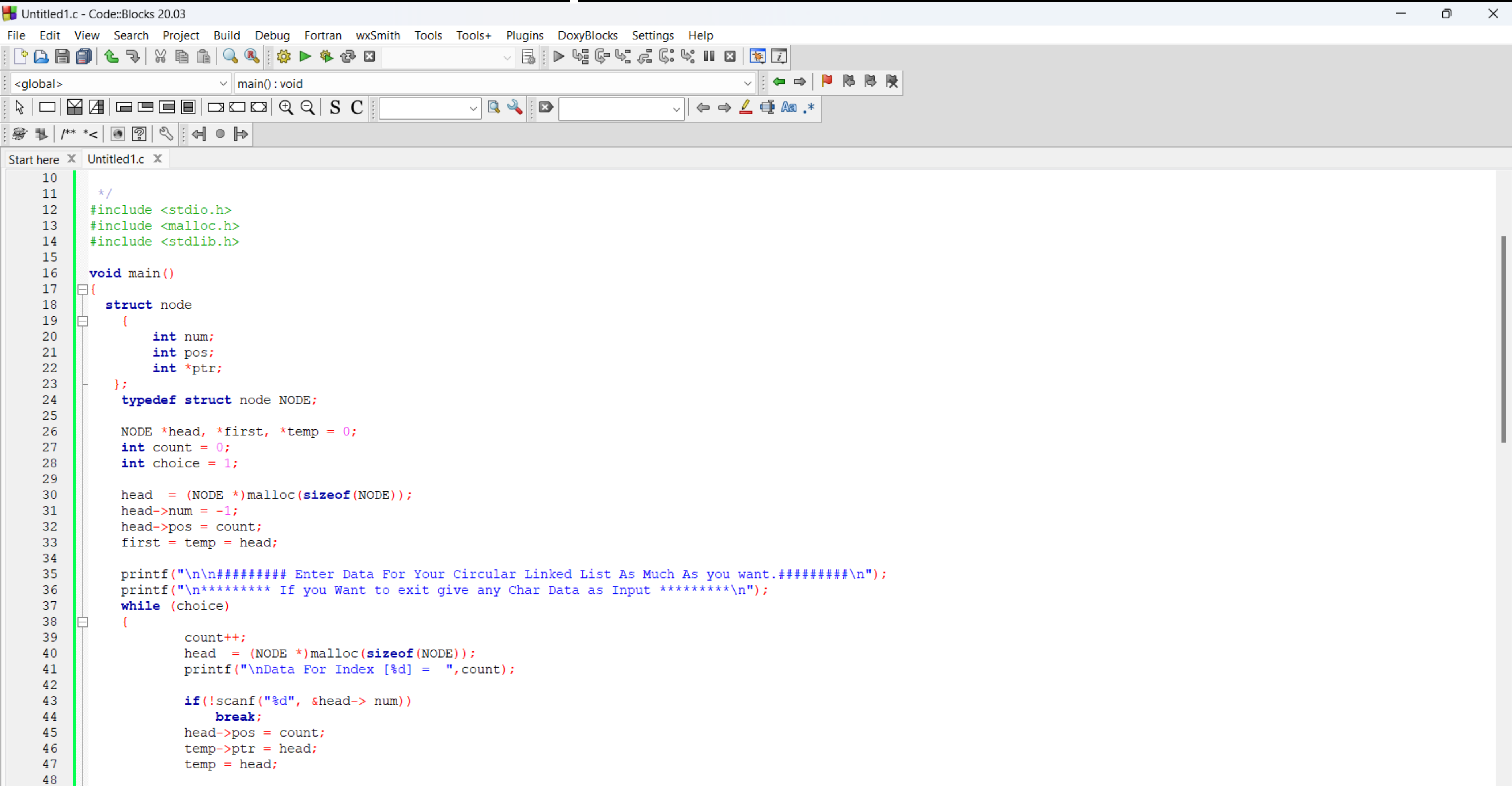


Circular Linked List with Sentinal value.



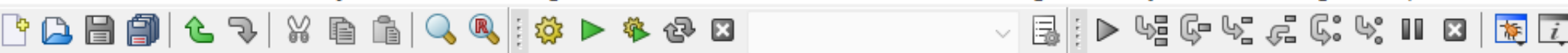
- **Header Node with Sentinel:** Assume that info part contains positive integers. Therefore the info part of a header node can be -1.

Implementation in C



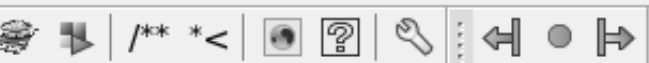
The screenshot shows the Code::Blocks 20.03 IDE with a C program for implementing a circular linked list. The code is written in a file named 'Untitled1.c'. The program includes standard headers for stdio, malloc, and stdlib. It defines a 'node' struct with 'num', 'pos', and 'ptr' fields. The 'main' function initializes a 'head' pointer, sets 'count' to 0, and 'choice' to 1. It then enters a loop where it prompts the user to enter data for the circular linked list. For each iteration, it allocates a new node, sets its 'num' value, updates 'pos' and 'ptr', and links it to the existing list. The loop continues until the user enters a character to exit.

```
10
11  */
12  #include <stdio.h>
13  #include <malloc.h>
14  #include <stdlib.h>
15
16  void main()
17  {
18      struct node
19      {
20          int num;
21          int pos;
22          int *ptr;
23      };
24      typedef struct node NODE;
25
26      NODE *head, *first, *temp = 0;
27      int count = 0;
28      int choice = 1;
29
30      head = (NODE *)malloc(sizeof(NODE));
31      head->num = -1;
32      head->pos = count;
33      first = temp = head;
34
35      printf("\n\n##### Enter Data For Your Circular Linked List As Much As you want.#####\n");
36      printf("\n***** If you Want to exit give any Char Data as Input *****\n");
37      while (choice)
38      {
39          count++;
40          head = (NODE *)malloc(sizeof(NODE));
41          printf("\nData For Index [%d] = ", count);
42
43          if (!scanf("%d", &head-> num))
44              break;
45          head->pos = count;
46          temp->ptr = head;
47          temp = head;
48
```



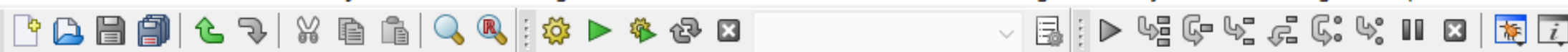
<global>

main() : void

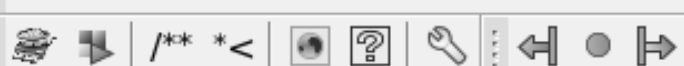


start here x Untitled1.c x

```
46         temp->ptr = head;
47         temp = head;
48
49         fflush(stdin);
50
51     }
52
53
54
55     temp->ptr = first;
56
57
58
59
60
61     printf("\n\n\n\n\n \t\t##### Printing Circular Linked List #####\n\n\n\n\n\n");
62     //Reset Value For printing Form First Node
63
64     temp = first;
65
66     while (1)
67     {
68         printf("\t\t[%d]   Address: %d   Data: %d   Next: %d \n", temp->pos, temp, temp->num, temp->ptr);
69
70         temp = temp->ptr;
71         if(temp->num == -1)
72             break;
73     }
74
75
76     printf("\n\n\n \t\t##### Deleted 2nd Element From Circular Linked List #####\n\n\n");
77
78     temp = first;
79     NODE *temp0;
80     while(1)
81     {
82
83         if(temp->pos == 2)
84         {
```

<global> main() : void



Start here x Untitled1.c x

```
70     temp = temp->ptr;
71     if(temp->num == -1)
72         break;
73 }
74
75
76 printf("\n\n\n \t\t##### Deleted 2nd Element From Circular Linked List #####\n\n");
77
78 temp = first;
79 NODE *temp0;
80 while(1)
81 {
82
83     if(temp->pos == 2)
84     {
85         temp0->ptr = temp->ptr;
86         break;
87     }
88
89     temp0 = temp;
90     temp = temp->ptr;
91
92 }
93 printf("\n\n\n \t\t##### Printing Circular Linked List #####\n\n\n\n\n");
94
95 temp = first;
96
97 while (1)
98 {
99     printf("\t\t[%d]    Address: %d    Data: %d    Next: %d \n", temp->pos, temp, temp->num, temp->ptr);
100
101     temp = temp->ptr;
102     if(temp->num == -1)
103         break;
104 }
105
106 }
107
```

Output

Untitled1.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global> main() : void

C:\Users\Hp\OneDrive\Desktop

Start here x Un

```
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
```

Enter Data For Your Circular Linked List As Much As you want.#####

***** If you Want to exit give any Char Data as Input *****

Data For Index [1] = 5

Data For Index [2] = 6

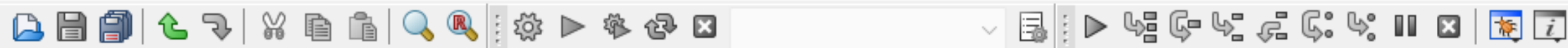
Data For Index [3] = 7

Data For Index [4] = 4

Data For Index [5] = x

Printing Circular Linked List

[0]	Address: 13308976	Data: -1	Next: 13309008
[1]	Address: 13309008	Data: 5	Next: 13309040
[2]	Address: 13309040	Data: 6	Next: 13309072
[3]	Address: 13309072	Data: 7	Next: 13309104



lobal> main() : void

t here x Un

t here x Un

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

```
[0] Address: 13308976 Data: -1 Next: 13309008
[1] Address: 13309008 Data: 5 Next: 13309040
[2] Address: 13309040 Data: 6 Next: 13309072
[3] Address: 13309072 Data: 7 Next: 13309104
[4] Address: 13309104 Data: 4 Next: 13308976
```

```
##### Deleted 2nd Element From Circular Linked List #####
```

```
##### Printing Circular Linked List #####
```

```
[0] Address: 13308976 Data: -1 Next: 13309008
[1] Address: 13309008 Data: 5 Next: 13309072
[3] Address: 13309072 Data: 7 Next: 13309104
[4] Address: 13309104 Data: 4 Next: 13308976
```

```
Process returned -1 (0xFFFFFFFF) execution time : 14.025 s
Press any key to continue.
```


THANK
YOU