

# VARUN BHARGAVA – 241010282

## DATA STRUCTURES TASK-7

### Task 01: Queue:

(<https://github.com/varunnnnb/dsa-sem3-iiitnr/blob/main/lab7/lab7-1.c>)

Use a queue to convert a given positive integer into its binary representation.

Steps:

1. Input a decimal number (e.g., 19).
2. Repeatedly divide the number by 2, store remainders in a queue.
3. Dequeue elements to print the binary equivalent.

The image displays two screenshots of a C++ IDE, likely Visual Studio Code, showing the implementation of a queue-based binary conversion algorithm. The top screenshot shows the initial code structure, including the queue definition and the main function. The bottom screenshot shows the completed code, including the enqueue and dequeue functions, and the main function logic for converting the decimal number 19 to its binary representation 10011.

```
lab7-1.c U X
lab7 > C lab7-1.c > main()
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define max 100
5
6 struct Queue
7 {
8     int data[max];
9     int front, rear;
10 };
11
12 void enqueue(struct Queue *x, int a)
13 {
14     if (x->rear < max - 1)
15         x->data[++(x->rear)] = a;
16 }
17
18 int dequeue(struct Queue *x)
19 {
20     if (x->front <= x->rear)
21         return x->data[(x->front)++];
22     return -1;
23 }
24
25 int isEmpty(struct Queue *x)
26 {
27     return (x->front > x->rear);
28 }
29
30 int main()
31 {
32     int n;
33     printf("Enter a number: ");
34     scanf("%d", &n);
35
36     struct Queue x;
37     x.front = 0;
38     x.rear = -1;
39
40     int temp = n;
41     if (temp == 0)
42     {
43         enqueue(&x, 0);
44     }
45     else
46     {
47         while (temp > 0)
48         {
49             enqueue(&x, temp % 2);
50             temp /= 2;
51         }
52     }
53
54     printf("The binary representation of %d is: ", n);
55     for (int i = x.rear; i >= x.front; i--)
56     {
57         printf("%d", x.data[i]);
58     }
59     printf("\n");
60
61     return 0;
62 }
```

PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa> cd "c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7" ; if (\$?) { gcc lab7-1.c -o lab7-1 } ; if (\$?) { .\lab7-1 }
Enter a number: 19
The binary representation of 19 is: 10011
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7>

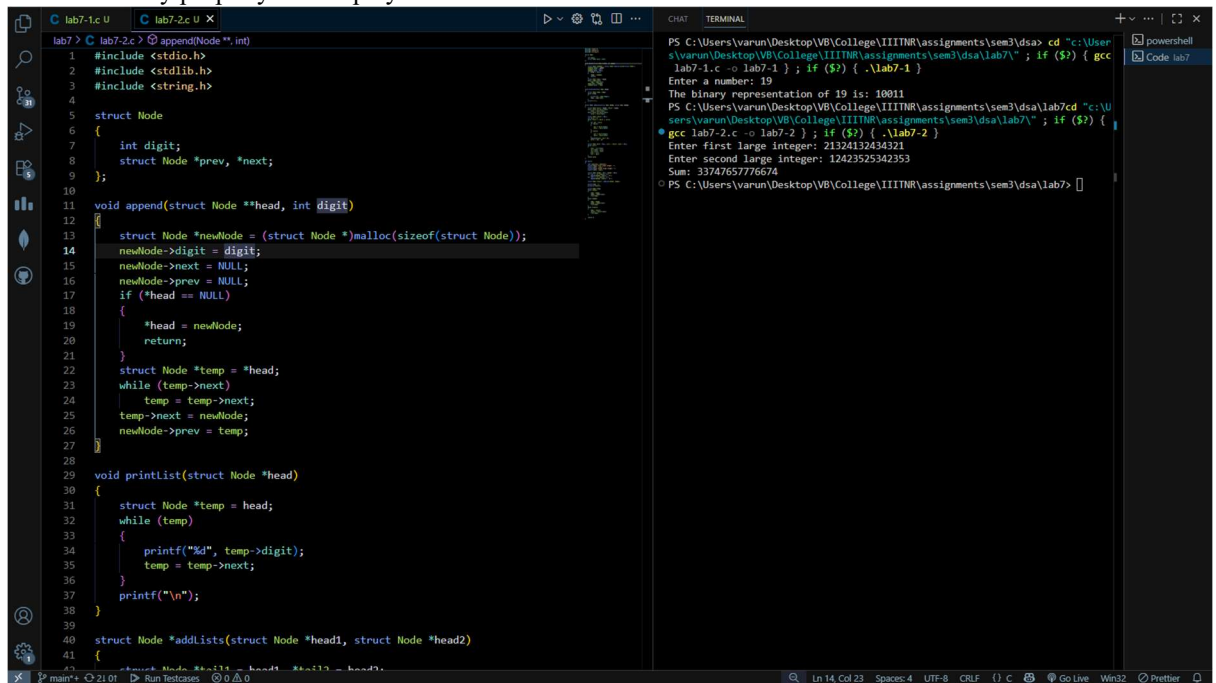
```
lab7-1.c U X
lab7 > C lab7-1.c > main()
25 int isEmpty(struct Queue *x)
29
30 int main()
31 {
32     int n;
33     printf("Enter a number: ");
34     scanf("%d", &n);
35
36     struct Queue x;
37     x.front = 0;
38     x.rear = -1;
39
40     int temp = n;
41     if (temp == 0)
42     {
43         enqueue(&x, 0);
44     }
45     else
46     {
47         while (temp > 0)
48         {
49             enqueue(&x, temp % 2);
50             temp /= 2;
51         }
52     }
53
54     printf("The binary representation of %d is: ", n);
55     for (int i = x.rear; i >= x.front; i--)
56     {
57         printf("%d", x.data[i]);
58     }
59     printf("\n");
60
61     return 0;
62 }
```

PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa> cd "c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7" ; if (\$?) { gcc lab7-1.c -o lab7-1 } ; if (\$?) { .\lab7-1 }
Enter a number: 19
The binary representation of 19 is: 10011
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7>

## Task 02: Doubly Linked List:

( <https://github.com/varunnrb/dsa-sem3-iiitnr/blob/main/lab7/lab7-2.c> )

1. Use a doubly linked list to add two large integers (each digit as one node).
2. Steps:
  1. Store each digit of the two numbers in separate doubly linked lists.
  2. Perform addition digit by digit from the rightmost (using backward traversal).
  3. Handle carry properly and display the result list.



```
lab7-2.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 struct Node
6 {
7     int digit;
8     struct Node *prev, *next;
9 };
10
11 void append(struct Node **head, int digit)
12 {
13     struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
14     newNode->digit = digit;
15     newNode->next = NULL;
16     newNode->prev = NULL;
17     if (*head == NULL)
18     {
19         *head = newNode;
20         return;
21     }
22     struct Node *temp = *head;
23     while (temp->next)
24         temp = temp->next;
25     temp->next = newNode;
26     newNode->prev = temp;
27 }
28
29 void printlist(struct Node *head)
30 {
31     struct Node *temp = head;
32     while (temp)
33     {
34         printf("%d", temp->digit);
35         temp = temp->next;
36     }
37     printf("\n");
38 }
39
40 struct Node *addlists(struct Node *head1, struct Node *head2)
41 {
42     // struct Node *result = head1;
43     // struct Node *result = head2;
```

```
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa> cd "c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7\" ; if ($?) { gcc lab7-1.c -o lab7-1 } ; if ($?) { .\lab7-1 }
Enter a number: 19
The Binary representation of 19 is: 10011
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7> cd "c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7\" ; if ($?) { gcc lab7-2.c -o lab7-2 } ; if ($?) { .\lab7-2 }
Enter first large integer: 21324132434321
Enter second large integer: 12423525342353
Sum: 33747657776674
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7>
```

```
lab7-1.c U lab7-2.c X
lab7-2.c > appendNode*,int)
40 struct Node *addLists(struct Node *head1, struct Node *head2)
41 {
42     struct Node *tail1 = head1, *tail2 = head2;
43     while (tail1 && tail1->next)
44         tail1 = tail1->next;
45     while (tail2 && tail2->next)
46         tail2 = tail2->next;
47
48     struct Node *result = NULL;
49     int carry = 0;
50     while (tail1 || tail2 || carry)
51     {
52         int sum = carry;
53         if (tail1)
54         {
55             sum += tail1->digit;
56             tail1 = tail1->prev;
57         }
58         if (tail2)
59         {
60             sum += tail2->digit;
61             tail2 = tail2->prev;
62         }
63         append(&result, sum % 10);
64         carry = sum / 10;
65     }
66
67     struct Node *prev = NULL, *curr = result, *next = NULL;
68     while (curr)
69     {
70         next = curr->next;
71         curr->next = prev;
72         curr->prev = next;
73         prev = curr;
74         curr = next;
75     }
76     return prev;
77 }
78
79 int main()
80 {
81     char num1[101], num2[101];
82     printf("Enter first large integer: ");
83     scanf("%100s", num1);
84     printf("Enter second large integer: ");
85     scanf("%100s", num2);
86
87     struct Node *head1 = NULL, *head2 = NULL;
88     for (int i = 0; num1[i]; i++)
89         append(&head1, num1[i] - '0');
90     for (int i = 0; num2[i]; i++)
91         append(&head2, num2[i] - '0');
92
93     struct Node *result = addLists(head1, head2);
94
95     printf("Sum: ");
96     printList(result);
97
98     struct Node *temp;
99     while (head1)
100     {
101         temp = head1;
102         head1 = head1->next;
103         free(temp);
104     }
105     while (head2)
106     {
107         temp = head2;
108         head2 = head2->next;
109         free(temp);
110     }
111     while (result)
112     {
113         temp = result;
114         result = result->next;
115         free(temp);
116     }
117
118     return 0;
119 }
```

```
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa> cd "c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7\" ; if ($?) { gcc lab7-1.c -o lab7-1 } ; if ($?) { .\lab7-1 }
Enter a number: 19
The binary representation of 19 is: 10011
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7cd "c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7\" ; if ($?) { gcc lab7-2.c -o lab7-2 } ; if ($?) { .\lab7-2 }
Enter first large integer: 21324132434321
Enter second large integer: 12423525342353
Sum: 33747657776674
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7>
```

## Task 03: Circular Doubly Linked List:

( <https://github.com/varunnnb/dsa-sem3-iiitnr/blob/main/lab7/lab7-3.c> )

Use a circular doubly linked list to represent and add two polynomials.

Steps:

1. Each node contains: coefficient, power, and links (prev, next).
2. Insert terms of the two polynomials.
3. Traverse circularly and add like terms (same power).
4. Print the resulting polynomial in decreasing order of power.

Example:

Polynomial A:  $3x^3 + 4x^2 + 2$  (in CDLL 1)

Polynomial B:  $5x^3 + 2x + 1$  (in CDLL 2)

Compare the power while traversing the list and add the sum in result list along

with power and display the final result.

Result:  $8x^3 + 4x^2 + 2x + 3$

```
lab7-3.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct Node
5 {
6     int coeff, power;
7     struct Node *prev, *next;
8 };
9
10 void insertTerm(struct Node **head, int coeff, int power)
11 {
12     struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
13     newNode->coeff = coeff;
14     newNode->power = power;
15     newNode->next = *head;
16     *head = newNode;
17 }
18
19 void printPoly(struct Node *head)
20 {
21     if (!head)
22         return;
23     struct Node *curr = head;
24     do
25     {
26         if (curr->power > 0)
27             printf("%d", curr->coeff);
28         curr = curr->next;
29     } while (curr != head);
30 }
31
32 struct Node *addPoly(struct Node *head1, struct Node *head2)
33 {
34     struct Node *result = NULL;
35     struct Node *p1 = head1, *p2 = head2;
36     if (!p1 || !p2)
37         return NULL;
38     do
39     {
40         int c1 = p1->coeff, p1p = p1->power;
41         int c2 = p2->coeff, p2p = p2->power;
42         if (p1p > p2p)
43             insertTerm(&result, c1, p1p);
44         else if (p2p > p1p)
45             insertTerm(&result, c2, p2p);
46         else
47             insertTerm(&result, c1 + c2, p1p);
48         if (p1 == head1)
49             p1 = NULL;
50         if (p2 == head2)
51             p2 = NULL;
52     } while (p1 || p2);
53     return result;
54 }
55
56 int main()
57 {
58     struct Node *head1 = NULL;
59     struct Node *head2 = NULL;
60     insertTerm(&head1, 3, 2);
61     insertTerm(&head1, 1, 1);
62     insertTerm(&head1, 5, 0);
63     insertTerm(&head2, 5, 3);
64     insertTerm(&head2, 3, 2);
65     insertTerm(&head2, 1, 1);
66     insertTerm(&head2, 10, 0);
67     printPoly(head1);
68     printPoly(head2);
69     struct Node *result = addPoly(head1, head2);
70     printPoly(result);
71     return 0;
72 }
```

```
lab7-3.h
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct Node
5 {
6     int coeff, power;
7     struct Node *prev, *next;
8 };
9
10 void insertTerm(struct Node **head, int coeff, int power);
11 void printPoly(struct Node *head);
12 struct Node *addPoly(struct Node *head1, struct Node *head2);
13
14 int main();
```

```
terminal
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7> gcc lab7-3.c -o lab7-3 ; if ($?) { .\lab7-3 }
Enter number of terms in Polynomial 1: 3
Enter coefficient and power for term 1 of Polynomial 1 (e.g. 3 2 for 3x^2): 3 2
Enter coefficient and power for term 2 of Polynomial 1 (e.g. 3 2 for 3x^2): 1 1
Enter coefficient and power for term 3 of Polynomial 1 (e.g. 3 2 for 3x^2): 5 0
Enter number of terms in Polynomial 2: 3
Enter coefficient and power for term 1 of Polynomial 2 (e.g. 5 3 for 5x^3): 5 3
Enter coefficient and power for term 2 of Polynomial 2 (e.g. 5 3 for 5x^3): 3 2
Enter coefficient and power for term 3 of Polynomial 2 (e.g. 5 3 for 5x^3): 1 1
Polynomial 1: 3x^2 + 1x^1 + 5
Polynomial 2: 5x^3 + 3x^2 + 1x^1 + 10
Result: 5x^3 + 8x^2 + 4x + 15
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7>
```



```
lab7-1.c U lab7-2.c U lab7-3.c U X
lab7-3.c > insertTerm(Node**int,int)
59 struct Node*addPoly(struct Node*head1, struct Node*head2)
60 do
61 if (p1 && (p2 || pow1 > pow2))
62 else if (p2 && (p1 || pow2 > pow1))
63 {
64 insertTerm(&result, c2, pow2);
65 p2 = p2->next;
66 if (p2 == head2)
67 p2 = NULL;
68 }
69 else if (p1 && p2 && pow1 == pow2)
70 {
71 insertTerm(&result, c1 + c2, pow1);
72 p1 = p1->next;
73 p2 = p2->next;
74 if (p1 == head1)
75 p1 = NULL;
76 if (p2 == head2)
77 p2 = NULL;
78 }
79 while (p1 || p2);
80 return result;
81 }
82
83 int main()
84 {
85 struct Node*poly1 = NULL,*poly2 = NULL;
86 int n1, n2, coeff, power;
87
88 printf("Enter number of terms in Polynomial 1: ");
89 scanf("%d", &n1);
90 for (int i = 0; i < n1; i++)
91 {
92 printf("Enter coefficient and power for term %d of Polynomial 1 (e.g. 5 3 for 5x^3): ");
93 scanf("%d %d", &coeff, &power);
94 insertTerm(&poly1, coeff, power);
95 }
96
97 printf("Enter number of terms in Polynomial 2: ");
98 scanf("%d", &n2);
99
100 for (int i = 0; i < n2; i++)
101 {
102 printf("Enter coefficient and power for term %d of Polynomial 2 (e.g. 5 3 for 5x^3): ");
103 scanf("%d %d", &coeff, &power);
104 insertTerm(&poly2, coeff, power);
105 }
106
107 struct Node*result = addPoly(poly1, poly2);
108 printf("Result: ");
109 printPoly(result);
110
111 struct Node*curr, *temp;
112 curr = poly1;
113 if (curr)
114 {
115 do
116 {
117 temp = curr;
118 curr = curr->next;
119 free(temp);
120 } while (curr != poly1);
121 }
122 curr = poly2;
123 if (curr)
124 {
125 do
126 {
127 temp = curr;
128 curr = curr->next;
129 free(temp);
130 } while (curr != poly2);
131 }
132 curr = result;
133 if (curr)
```

```
over for term 11496 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11497 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11498 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11499 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11500 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11501 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11502 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11503 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11504 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11505 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11506 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11507 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11508 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11509 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11510 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11511 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11512 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11513 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7> cd "c:\Us
ers\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7\" ; if ($?) { gcc
lab7-3.c -o lab7-3 } ; if ($?) { .\lab7-3 }
Enter number of terms in Polynomial 1: 3
Enter coefficient and power for term 1 of Polynomial 1 (e.g. 3 2 for 3x^2): 3 2
Enter coefficient and power for term 2 of Polynomial 1 (e.g. 3 2 for 3x^2): 1 1
Enter coefficient and power for term 3 of Polynomial 1 (e.g. 3 2 for 3x^2): 5 0
Enter number of terms in Polynomial 2: 3
Enter coefficient and power for term 1 of Polynomial 2 (e.g. 5 3 for 5x^3): 1 2
Enter coefficient and power for term 2 of Polynomial 2 (e.g. 5 3 for 5x^3): 5 1
Enter coefficient and power for term 3 of Polynomial 2 (e.g. 5 3 for 5x^3): 5 0
Polynomial 1: 3x^2 + 1x^1 + 5
Polynomial 2: 1x^2 + 5x^1 + 5
Result: 4x^2 + 6x^1 + 10
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7>
```

```
lab7 > C lab7-1.c U C lab7-2.c U C lab7-3.c U X
lab7 > C lab7-3.c > insertTerm(Node **, int, int)
100 int main()
134 {
135     if (curr)
136     {
137         do
138         {
139             temp = curr;
140             curr = curr->next;
141             free(temp);
142         } while (curr != poly1);
143     }
144     curr = poly2;
145     if (curr)
146     {
147         do
148         {
149             temp = curr;
150             curr = curr->next;
151             free(temp);
152         } while (curr != poly2);
153     }
154     curr = result;
155     if (curr)
156     {
157         do
158         {
159             temp = curr;
160             curr = curr->next;
161             free(temp);
162         } while (curr != result);
163     }
164     return 0;
165 }
```

```
over for term 11496 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11497 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11498 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11499 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11500 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11501 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11502 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11503 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11504 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11505 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11506 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11507 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11508 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11509 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11510 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11511 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11512 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
over for term 11513 of Polynomial 2 (e.g. 5 3 for 5x^3): Enter coefficient and p
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7> cd "c:\Us
ers\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7" ; if ($?) { gcc
lab7-3.c -o lab7-3 } ; if ($?) { .\lab7-3 }
Enter number of terms in Polynomial 1: 3
Enter coefficient and power for term 1 of Polynomial 1 (e.g. 3 2 for 3x^2): 3 2
Enter coefficient and power for term 2 of Polynomial 1 (e.g. 3 2 for 3x^2): 1 1
Enter coefficient and power for term 3 of Polynomial 1 (e.g. 3 2 for 3x^2): 5 0
Enter number of terms in Polynomial 2: 3
Enter coefficient and power for term 1 of Polynomial 2 (e.g. 5 3 for 5x^3): 1 2
Enter coefficient and power for term 2 of Polynomial 2 (e.g. 5 3 for 5x^3): 5 1
Enter coefficient and power for term 3 of Polynomial 2 (e.g. 5 3 for 5x^3): 5 0
Polynomial 1: 3x^2 + 1x^1 + 5
Polynomial 2: 1x^2 + 5x^1 + 5
Result: 4x^2 + 6x^1 + 10
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab7>
```