```
#include <iostream>
using namespace std;
void swap(int *xp,int *yp)
{
int temp =*xp;
*xp=*yp;
*yp=temp;
}
//A function to implement bubble sort
void bubbleSort(int arr[],int n){
        int i,j;
        for(i=0; i<n-i; i++)
        //last i elements are already in place
        for(j=0;j<n-i-1;j++)
                if(arr[j] > arr[j+1])
                swap(&arr[j],&arr[j+1]);
}
//Function to print an array
void printArray(int arr[],int size)
{
int i;
  for(i=0;i<size;i++)
  cout<<arr[i]<<" ";
  cout<<endl;
}
int main() {
        int arr[]={64,34,25,12,22,11,90};
        int n = sizeof(arr)/sizeof(arr[0]);
        bubbleSort(arr,n);
        cout<<"Sorted array: \n";
        printArray(arr,n);
        return 0;
}
```

Original Array
arr[]=

64	34	25	12	22	11	90

First pass
i=0

J=0

34	64	25	12	22	11	90

64>34, true then swap

J=1

34	25	64	12	22	11	90

64>25, true then swap

J=2

34	25	12	64	22	11	90

64>12, true then swap

J=3

34	25	12	22	64	11	90

64>22, true then swap

J=4

34	25	12	22	11	64	90

64>11, true then swap

J=5

34	25	12	22	11	64	90

64>90, false then no swap

arr[]=

34	25	12	22	11	64	90

second pass

i=1

J=0

25	34	12	22	11	64	90

34>25, true then swap

J=1

25	12	34	22	11	64	90

34>12, true then swap

J=2

25	12	22	34	11	64	90

34>22, true then swap

J=3

25	12	22	11	34	64	90

34>11, true then swap

J=4

25	12	22	11	34	64	90

34>64, false then no swap

arr[]=

25	12	22	11	34	64	90

---

third pass

i=2

J=0

12	25	22	11	34	64	90

25>12, true then swap

J=1

12	22	25	11	34	64	90

25>22, true then swap

J=2

12	22	11	25	34	64	90

25>11, true then swap

J=3

12	22	11	25	34	64	90

25>34, false then no swap

arr[]=

12	22	11	25	34	64	90

---

fourth pass

i=3

J=0

12	22	11	25	34	64	90

12>22, false then no swap

J=1

12	11	22	25	34	64	90

22>11, true then swap

J=2

12	11	22	25	34	64	90

22>25, false then no swap

After fourth pass array will be as follows

arr[]=

12	11	22	25	34	64	90

```cpp
#include <iostream>
using namespace std;
void swap(int *xp,int *yp)
{
int temp =*xp;
*xp=*yp;
*yp=temp;
}
//A function to implement bubble sort
void bubbleSort(int arr[],int n){
        int i,j;
        for(i=0; i<n-i; i++)
        //last i elements are already in place
        for(j=0;j<n-i-1;j++)
                if(arr[j] < arr[j+1])
                swap(&arr[j],&arr[j+1]);
}
//Function to print an array
void printArray(int arr[],int size)
{
int i;
  for(i=0;i<size;i++)
  cout<<arr[i]<<" ";
  cout<<endl;
}
int main() {
        int arr[]={64,34,25,12,22,11,90};
        int n = sizeof(arr)/sizeof(arr[0]);
        bubbleSort(arr,n);
        cout<<"Sorted array: \n";
        printArray(arr,n);
        return 0;
}
```

Original Array
arr[]=

64	34	25	12	22	11	90

First pass
i=0

	64	34	25	12	22	11	90
J=0	64	34	25	12	22	11	90
J=1	64	34	25	12	22	11	90
J=2	64	34	25	12	22	11	90
J=3	64	34	25	22	12	11	90
J=4	64	34	25	22	12	11	90
J=5	64	34	25	22	12	90	11

64<34, false then no swap

34<25, false then no swap

25<12, false then no swap

12<22, true then swap

12<11, false then no swap

11<90, true then swap

arr[]=

64	34	25	22	12	90	11

second pass
i=1

64<34, false then no swap

J=0

64	34	25	22	12	90	11

34<25, false then no swap

J=1

64	34	25	22	12	90	11

25<22, false then no swap

J=2

64	34	25	22	12	90	11

22<12, false then no swap

J=3

64	34	25	22	12	90	11

12<90, true then swap

J=4

64	34	25	22	90	12	11

arr[]=

64	34	25	22	90	12	11

third pass
i=2

64<34, false then no swap

J=0

64	34	25	22	90	12	11

34<25, false then no swap

J=1

64	34	25	22	90	12	11

25<22, false then no swap

J=2

64	34	25	22	90	12	11

22<90, true then swap

J=3

64	34	25	90	22	12	11

arr[]=

64	34	25	90	22	12	11

fourth pass
i=3

64<34, false then no swap

J=0

64	34	25	90	22	12	11

34<25, false then no swap

J=1

64	34	25	90	22	12	11

25<90, true then swap

J=2

64	34	90	25	22	12	11

After fourth pass array will be as follows
arr[]=

64	34	90	25	22	12	11

Question 3
(a) arr[]={6,5,12,10,9,1}

0	1	2	3	4	5
6	5	12	10	9	1

6	5	12

10	9	1

6	5

12

10

9	1

6

5

1

9

5	6

12

10

1	9

5	6	12

1	9	10

1	5	6	9	10	12
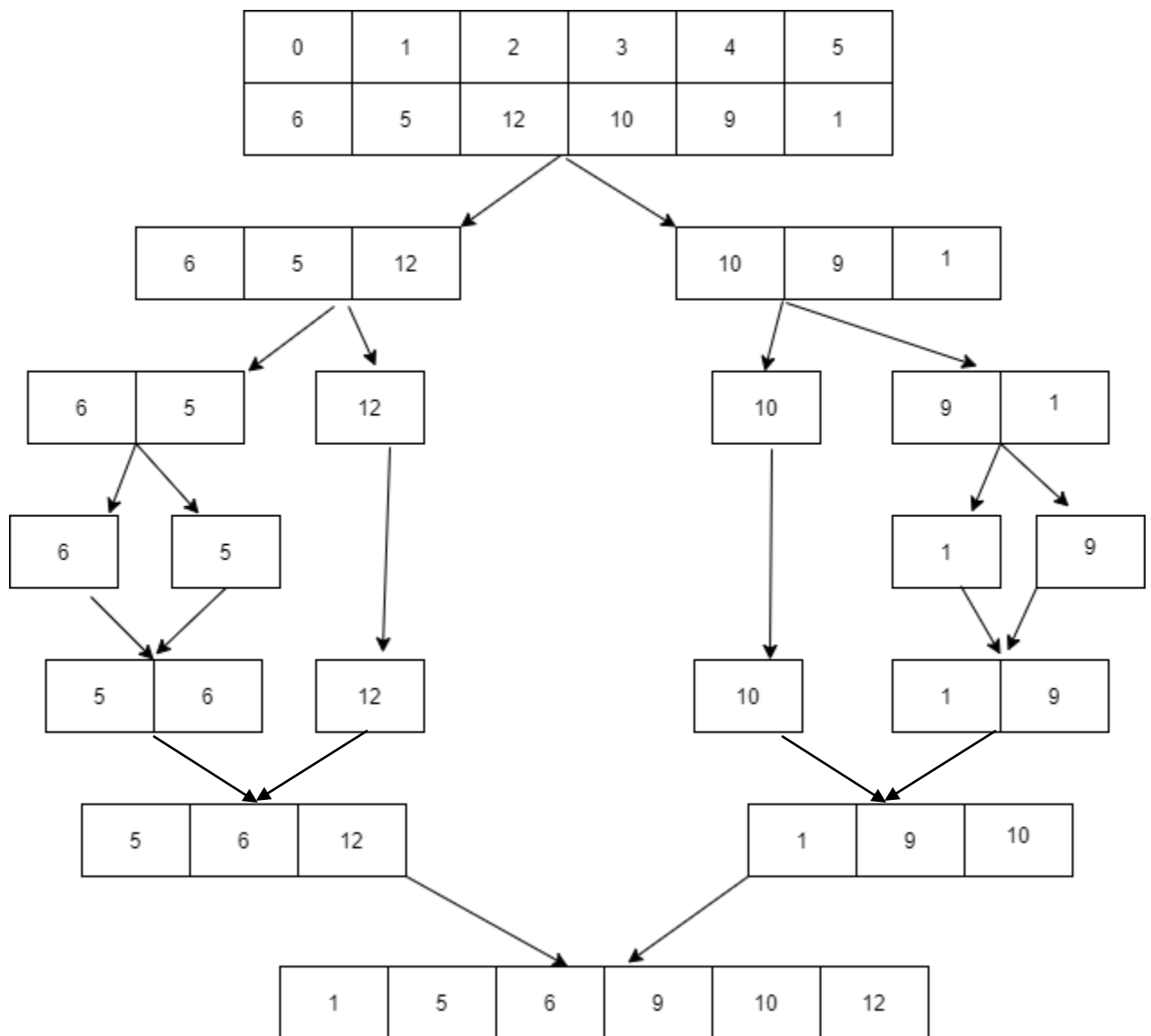
Question 3
(b) arr[]={10,9,1,6,5,12}

0	1	2	3	4	5
10	9	1	6	5	12

10	9	1

6	5	12

10	9

1

6

5	12

10

9

5	12

9	10

1

6

5	12

1	9	10

5	6	12

1	5	6	9	10	12
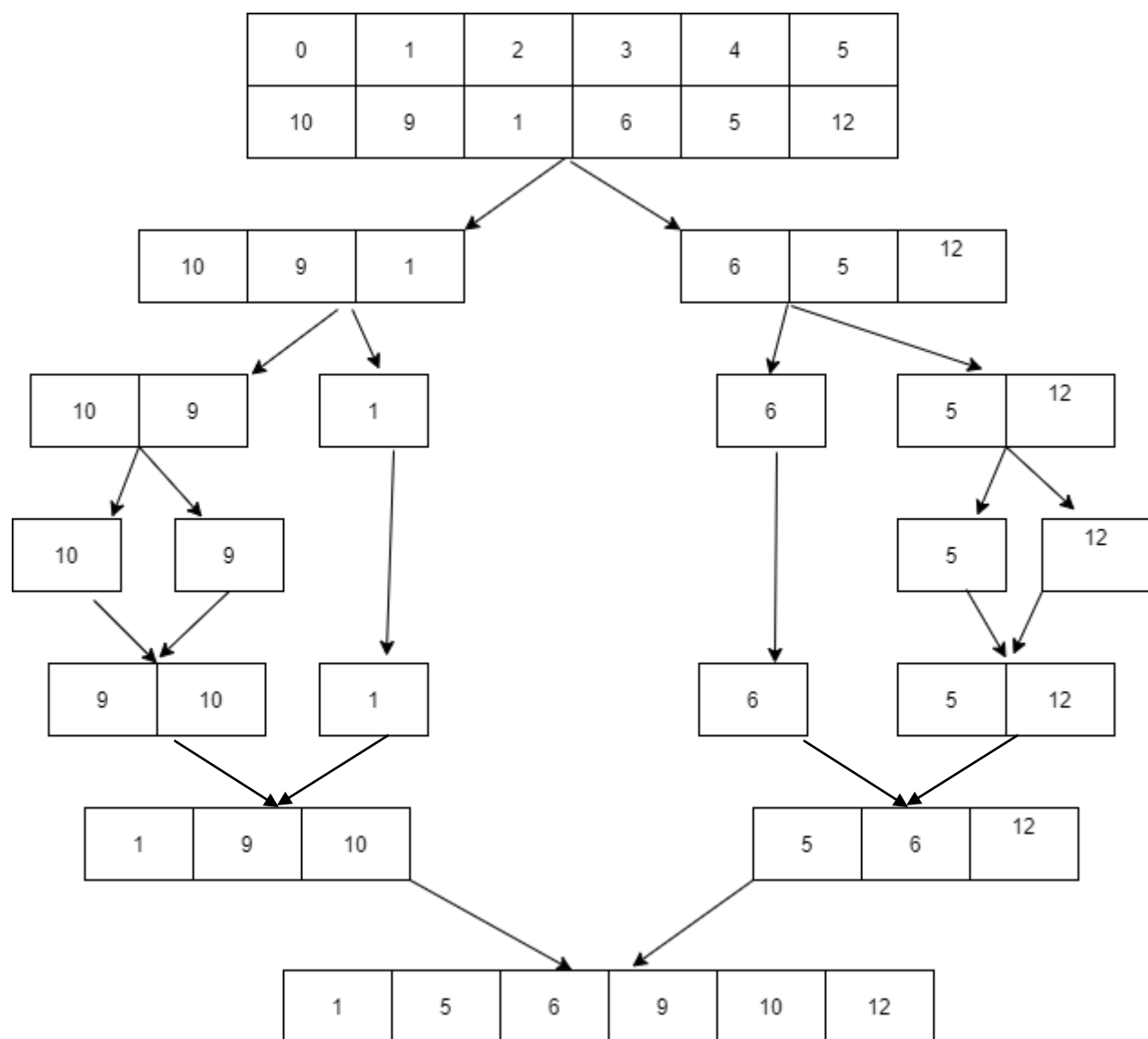
Question 4
Convert infix to postfix 2 * 3 /(2 - 1) + 5 * 3

Expression	Stack	Output (postfix expression)
2		2
*	*	2
3	*	23
/	/	23*
(	(/	23*
2	(/	23*2
-	-(/	23*2
1	-(/	23*21
)	/	23*21-
+	+	23*21-/
5	+	23*21-/5
*	*+	23*21-/5
3	*+	23*21-/53
		23*21-/53*+

Question 5
Evaluate 623 + - 382 / + * 2 ^ 3 +

Scan from left to right

6,2,3,+ : add 2 and 3, then continue to scan

6,5,- : subtract 5 from 6, then continue to scan

1,3,8,2,/ : divide 8 by 2, then continue to scan

1,3,4,+ : add 3 and 4, then continue to scan

1,7,* : multiply 1 by 7, then continue to scan

7,2,^ : rise 7 by 2, then continue to scan

49,3,+ : add 49 and 3

52