Write the output of the following programs (if any). If there is an error in the program, correct the code and then write the output.

Tip: Use python tutor (https://pythontutor.com/visualize.html#mode=edit) for line by line execution of programs for a better understanding, first try to solve by yourself.

| ```cpp
#include<iostream>
using namespace std;
int x;

void fun1() {
    int x = 7;
    static int y = 10;
    x++;
    y++;
    cout << y << " " << ::x << endl;
}
void fun2() {
    int x = 27;
    int y = 20;
    ::x++; x++;
    y++;
    cout << x << " " << y << " " << ::x << endl;
}
int main() {
    fun1();
    fun2();
    fun1();
    fun2();
    return 0;
}
``` | |
| ```cpp
int list[5] = { 2,4,8,10,-1 };
int nextList[5] = { 3,-1,0,1,-1 };
int start = 2;
int Free = 4;
void magic(int val, int position) {
        int start = ::start;
        for (int i = 0; i < position - 1; i++)
                start = nextList[start];
        list[Free] = val; nextList[Free] = nextList[start];
        nextList[start] = Free++;
}
void magic() {
        int start = ::start;
        while (start != -1) {
                cout << list[start] << "->";
                start = nextList[start];
        }
        cout << "*" << endl;
}
int main()
{
        magic();
        magic(5, 2);
        magic();
        return 0;
}
``` | |
| ```cpp
float calc(int y, int x) {
``` | |

```cpp
        return (y + x + 7.0 / 2);
}
int main()
{
        float i = 9.5;
        int j = 4.5;
        cout << calc(i, j) << endl;

}
```

```cpp
#include <iostream>
using namespace std;
int main()
{
        int arr[] = { 1, 10, 3, 7, 5, 6, 9, 2, 8 };
        const int N = sizeof(arr) / sizeof(arr[0]);
        int i;
        int temp[N + 1];

        for (int i = 0; i <= N; i++) {
                temp[i] = 0;
        }

        for (i = 0; i < N; i++) {
                temp[arr[i] - 1] = 1;
        }

        int ans;
        for (i = 0; i <= N; i++) {
                if (temp[i] == 0)
                        ans = i + 1;
        }

        ans &= ans;
        cout << ans << endl;
}
```

```cpp
int mystery(int x, int n)
{
        int val;
        val = 1;
        if (n >= 0)
        {
                if (n % 3 > 1)
                        val = val * x;
                else
                        val = val * 2;
        }
        return val;
}
int main()
{
        cout << "The mysterious value is: " << mystery(5, 2);
}
```

```cpp
void calc(float&, int&, int&);
int main()
{
        int val1 = 2, val2 = 3;
        float res;
        calc(res, val1, val2);
        cout << val1 << " " << val2 << endl;
        cout << "Reult is : " << res;
```

```cpp
        return 0;
}
void calc(float& r, int& para1, int& para2)
{
        para1 = 4 + para2 * 2;
        para2 += 3 * para1;
        r = para1 + para2 / 2.0;
}
```

```cpp
int mystery(int value)
{
        static int count = 3;
        value += count;
        return value;
}
int main()
{
        int val1;
        for (int c = 0; c <= 10; c += 3)
        {
                val1 = mystery(c);
                cout << val1 << endl;
        }
        return 0;
}
```

```cpp
void fun1(int&);
void fun2(int&);
int fun3(int);
int main()
{
        int val = 3;
        fun1(val);
        cout << val;
        return 0;
}
void fun1(int& p1)
{
        p1++;
        fun2(p1);
        p1++;
}
void fun2(int& p2)
{
        p2 = fun3(p2);
        p2++;
}
int fun3(int p3)
{
        p3 = p3 * 3;
        return p3;
}
```

```cpp
int Quad(int n)
{
        return (n * n * n * n);
}
int main() {
        int num = 1634;
        int res = 0;
        int remainder;
```

```cpp
        int n = num;
        while (n != 0)
        {
                remainder = n % 10;
                res = res + Quad(remainder);
                n = n / 10;
        }
        cout << "\n Result " << res;
        return 0;
}
```

```cpp
int WHAT(int A[], int N) {
        int ANS = 0;
        int S = 0;
        int E = N - 1;
        for (S = 0, E = N - 1; S < E; S++, E--)
                ANS += A[S] - A[E];
        return ANS;
}
int main() {
        int A[] = { 1, 2, 3, 4, -5, 1, 3, 2, 1 };
        cout << WHAT(A, 7);
        return 0;
}
```

```cpp
int get(int N = 0)
{
        int static x = 0;
        return x++;
}
int main()
{
        const int N = 6;
        int nums[] = { 1,2,3,4,5,6 };
        int idx = 1;
        while (idx)
        {
                idx = get(get());
                if (idx >= N)
                {
                        break;
                }
                cout << nums[idx] << endl;
        }
        return 0;
}
```

```cpp
void Sum(int a) {
        cout << a + 100 << endl;
}
void Sum(int a, int b, int c = 10) {
        cout << a + b + c << endl;
}
int main() {
        Sum('A');
        Sum('B', 30);
        Sum(20, 30, 90.5);
        return 0;
}
```

```cpp
#include<iostream>
using namespace std;
```

```cpp
bool Its_Magic(int arr[][4], int rows, int cols) {
        if (rows != cols)
                return false;

        for (int i = 0, s = i; i < rows; ++i, s++) {
                if (cout << arr[arr[i][i]][arr[0][0]] <<
           "\n" && arr[0][0] != arr[i][i])
                        return false;

                for (int j=i+1; j<cols; ++j, cout<<arr[j-1][i])
        {
                        if (arr[i][j] != arr[j][i] && cout<<s--)
                                return false;
                }
        }
        return true;
}

int main()
{
        int arr[4][4] = { 3, 2, 3, 8, 2, 3, 7, 9, 3, 7, 3, 2,
                8, 9, 2, 3 };

        if (Its_Magic(arr, 4, 4))
                std::cout << "Abracadabra";
}
```

```cpp
int fun(int x) {
        return x % 3 + 1;
}
int main()
{
        int b = 5;
        int y = 2 + fun(3 * b + 1);
        int z = fun(fun(y));
        cout << y << "-" << z;
}
```

```cpp
#include<iostream>
using namespace std;
int function(int x) {
        cout << "int";
}
int function(float x) {
        cout << "float";
}
int main() {
        double x;
        function(x);
}
```

```cpp
void fun4()
{
        cout << "-";
}
void fun3() {
        cout << "+";
        fun4();
        cout << "+";
}
void fun2() {
        cout << "/";
```

```cpp
        fun3();
        cout << "/";
}
void fun1() {
        cout << "*";
        fun2();
        cout << "*";


}
int main() {
        fun1();
        return 0;
}
```

```cpp
#include<iostream>
using namespace std;
void mystery1(int array1[], int n1, int& index);
void mystery2(int array2[], int a2[], int n2, int index2);
void mystery3(int array3[], int n3, int index3);
void mystery4(int array4[], int array3[], int index3);
int main() {
        int k = 3;
        int arr[5] = { 1, 2, 3, 4, 5 };
        mystery1(arr, 5, k);
        for (int i = 0; i < 5; i++)
                cout << arr[i] << " ";
        cout << "\n Value of k is " << k << endl;
        return 0;
}
void mystery1(int array1[], int n1, int& index) {
        const int MAXOFFSET = 100;
        int temp[MAXOFFSET];
        if (index > 0)
        {
                mystery2(temp, array1, n1, index);
                mystery3(array1, n1, index);
                mystery4(temp, array1, index);
        }
}
void mystery2(int array2[], int a2[], int n2, int index2) {
        for (int j = 0; j < index2; j++)
                array2[j] = a2[n2 - index2 + j];


}
void mystery3(int array3[], int n3, int index3) {
        for (int i = n3 - 1; i >= index3; i--)
        {
                array3[i] = array3[i - index3];
                index3++;
        }
}
void mystery4(int array4[], int array3[], int index3) {
        for (int i = 0; i < index3; i++)
                array3[i] = array4[i];
}
```

```cpp
int main() {
        int nrows = 3, ncols = 4;
        int A[2][3][4] = { {
        { 1, 3, 2 },
        { 4, 5 },
```

```cpp
                { 7, 8, 9 } },
                { { 4 },
                { 5, 5, 7 },
                { -2, 3, 4 } }
        };

        int b[2][4] = { { 0 } };

        for (int i = 0; i < 2; ++i) {
                for (int j = 0; j < ncols; ++j)
                        for (int k = 0; k < nrows; ++k)
                                b[i][j] += A[i][k][j];
        }


        for (int i = 0; i < 2; ++i) {
                for (int j = 0; j < nrows; ++j) {
                        for (int k = 0; k < ncols; ++k)
                                cout << A[i][j][k] << " ";
                        cout << endl;
                }

                for (int j = 0; j < ncols; ++j)
                        cout << b[i][j] << " ";
                cout << endl;
        }

        return 0;
}
```

```cpp
void function(int[][3]);
int main()
{
        int a[3][3] = { { 1,2,3} , { 4,5,6} , {7,8,9} };
        function(a);
        cout << a[2][1];
        return 0;
}
void function(int b[][3])
{
        ++b;
        b[1][1] = 9;
}
```

```cpp
void e(int);
int main()
{
        int a;
        a = 3;
        e(a);
        e(a);
}
void e(static int n)
{
        if (n > 0)
                cout << n << " ";
        n--;
}
```

```cpp
void find(int a, int& b, int& c);
int main()
{
```

```cpp
        int one, two, three;
        one = 5;
        two = 10;
        three = 15;
        find(one, two, three);
        cout << one << ", " << two << ", " << three << endl;
        find(two, one, three);
        cout << one << ", " << two << ", " << three << endl;
        find(three, two, one);
        cout << one << ", " << two << ", " << three << endl;
        find(two, three, one);
        cout << one << ", " << two << ", " << three << endl;
}
void find(int a, int& b, int& c)
{
        int temp;
        c = a + 2 * b;
        temp = b;
        b = a;
        a = 2 * temp;
}
```

```cpp
void e(int n)
{
        static int x = 3;
        if (n > 0)
                cout << n << " " << x << endl;
        n--;
        x--;
}
int main()
{
        int a;
        a = 10;

        e(a--);
        e(a--);
        e(a--);
}
```

```cpp
void fun3(int& a)
{
        a++;
        cout << a;
}
void fun2(int& a) {
        fun3(++a);
        cout << a;
}
void fun1(int& a) {
        fun2(++a);
        cout << a;
}
int a = 5;
int main() {
        int a = 1;
        fun1(a);
        cout << a;
        return 0;
}
```

```cpp
int hello(int a, int& b, int& c) {
        int x;
        b *= 10;
        x = a * b;
        a += 2;
        c++;
        x -= c;
        return x;
}
int main() {
        int a = 10, b = 11, c = 12, result;
        cout << a << " " << b << " " << c << endl;
        result = hello(a, b, c);
        cout << result << endl;
        return 0;
}
```

```cpp
#include<iostream>
using namespace std;
void do_it(int& var)
{
        for (int i = 0; i < var; i++)
        {
                if (i % 2)
                        var -= 1;
                if (!i % 2)
                        var += 2;
                if (!2 + !!!!0 + !i)
                        var -= 1;
        }
}

void onceMore(int& var)
{
        int s = var;
        while (var -= 2, s--)
                cout << var + s << " ";

        s = var + 14;

        while (var += 2, s--)
                cout << var - s << " ";
}


int main() {
        int a = 10, & p = a, b = 8;
        do_it(p);
        cout << a << b << p<<endl;
        p = b;
        onceMore(p);
        cout << endl << a << b << p;
}
```

Q. Consider the C++ code given below and answer the given questions.

```cpp
#include <iostream>
using namespace std;

int main() {
        const int N = 3;
        int A[N] = { 3,2,1 };
        int B[N] = { 0 };

        for (int i = 0; i < N; ++i)
        {
                int length = 1;
                while (A[i] != 1) {
                        if (A[i] % 2)
                                A[i] = A[i] * 3 + 1;
                        else
                                A[i] /= 2;
                        ++length;
                }
                B[i] = length;
        }
        return 0;
}
```

a.   **What are the contents of array B when 1ˢᵗ iteration of for loop terminates.**

B[0]	B[1]	B[2]

b. **What are the contents of array B when 2ⁿᵈ iteration of for loop terminates.**

B[0]	B[1]	B[2]

c. **What are the contents of array B when 3ʳᵈ iteration of for loop terminates.**

B[0]	B[1]	B[2]

Q. Consider the C++ code given below and answer the given questions.

```cpp
int get(int N)
{
        static int i = 0;
        return N - (i++) - 1;
}
int main()
{
        int SIZE = 10;
        int arr[] = { 5,6,7,8,9,10,11,12,13,14 };
        int j;
        for (int i = 0; i < SIZE; i++)
        {
                j = get(SIZE);
                if (j == i)
                        continue;
                arr[i] += arr[j];
                arr[j] = arr[i] - arr[j];
                arr[i] -= arr[j];
        }
        return 0;
}
```

a. **What are the contents of array after 2nd iteration of for loop?**

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]	arr[7]	arr[8]	arr[9]
14	13	7	8	9	10	11	12	6	5

b. **What are the contents of array after 4th iteration of for loop?**

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]	arr[7]	arr[8]	arr[9]
14	13	12	11	9	10	8	7	6	5

c. **What are the contents of array after loop terminates?**

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]	arr[7]	arr[8]	arr[9]
5	6	7	8	9	10	11	12	13	14

Q. Consider the C++ code given below and answer the given questions.

```cpp
#include <iostream>
using namespace std;

void Abracadabra(int arr[], int n) {
   for (int i = 1; i < n; i++) {

      int a = arr[i];
      int b = i - 1;

      while (b >= 0 && arr[b] > a) {
         arr[b + 1] = arr[b];
         b--;
      }
      arr[b + 1] = a;
   }
}

int main() {
   int arr[] = { 55, 4, 11, 3, 2, 12, 45, 5, 31, 1};
   int n = sizeof(arr) / sizeof(arr[0]);

   Abracadabra(arr, n);
}
```

a. **What are the contents of array after 1st iteration of for loop?**

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]	arr[7]	arr[8]	arr[9]
4	55	11	3	2	12	45	5	31	1

b. **What are the contents of array after 6th iteration of for loop?**

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]	arr[7]	arr[8]	arr[9]
2	3	4	11	12	45	55	5	31	1

c. **What are the contents of array after for loop terminates?**

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]	arr[7]	arr[8]	arr[9]
1	2	3	4	5	11	12	31	45	55

Q. Consider the C++ code given below and answer the given questions.

```cpp
#include <iostream>
using namespace std;

void Gnome(int arr[], int n)
{
    int idx = 0, counter = 0;

    while (idx < n) {
        if (idx == 0)
            idx++;
        if (arr[idx] >= arr[idx - 1])
            idx++;
        else {
            int temp = arr[idx];
            arr[idx] = arr[idx - 1];
            arr[idx - 1] = temp;
            idx--;
        }
        counter++;
    }
}
int main()
{
    int arr[] = { 34, 2, 1, -9 };
    Gnome(arr, 4);
}
```

a. **What are the contents of array when value of counter is 4?**

arr[1]	arr[2]	arr[3]	arr[4]
1	2	34	-9

b. **What are the contents of array when value of counter is 9?**

arr[1]	arr[2]	arr[3]	arr[4]
-9	1	2	34

c. **What are the contents of array when value of counter is 11?**

arr[1]	arr[2]	arr[3]	arr[4]
-9	1	2	34

Q. Consider the C++ code given below and answer the given questions. There are NO Errors.

```cpp
#include<iostream>
using namespace std;

int main()
{
        int arr[10] = { 4, 3, 12, 31, 44, 32, 2, 69, 21, 11 };
        int sArr[5] = { 'P', 'F','T','S', 'f' };
        for (int i = 0; i < 10; i++)
        {
                if (!!!i)
                        arr[i] = sArr[i * 2] / 20;
                else if (arr[i] % 2)
                        arr[i - 1] = sArr[10 % i] - arr[7 % sArr[0] / 10];
                else
                        arr[i + 1] += arr[i - 1] + sArr[2 * i - i - 5] % 11;

                if ((sArr[1] == i + 11) && !(i % 3) && i)
                        arr[i] = arr[i - 3] + arr[i];
        }
}
```

a. **What are the contents of array after 4ᵗʰ iteration of for loop?**

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]	arr[7]	arr[8]	arr[9]

b. **What are the contents of array after 7ᵗʰ iteration of for loop?**

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]	arr[7]	arr[8]	arr[9]

c. **What are the contents of array after 5ᵗʰ iteration of loop?**

sArr[0]	sArr[1]	sArr[2]	sArr[3]	sArr[4]

d. **What are the contents of array after for loop terminates?**

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]	arr[5]	arr[6]	arr[7]	arr[8]	arr[9]