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.

```
void mystery(int* ptr, int s)
{
        ptr = new int[s];
        for (int i = 0, j = s; i < s; ++i, j--)
                *(ptr + i) = j;
}

int main()
{
        int* ptr, s = 5;
        mystery(ptr, s);
        for (int i = 0; i < s; ++i)
                cout << ptr[i] << " ";
        delete[] ptr;
        ptr = NULL;
        return 0;
}
```

```
const char* c[] = {"PF", "Exam" ,"PFMID-1","MID" };
char const** cp[] = { c + 2, c + 3, c , c + 1 };
char const*** cpp = cp;

int main()
{
        cout << *cpp[1] << endl;
        cout << *(*(*(cpp + 2) + 2) + 3) << endl;
        cout << (*cpp)[-1] << endl;
        cout << *(cpp + 3)[-1] << endl;

}
```

```

int main()
{
        const char* str[] = { "AAAAA", "BBBBB",
                "CCCCC", "DDDDD" };

        const char** sptr[] = { str + 3, str + 2,
                str + 1, str };

        const char*** pp;
        pp = sptr;
        ++pp;
        cout << **++pp + 2;

}
```

```

void f1(int*, int);
void f2(int*, int);
```

```
int main()
{
        int a;
        int b;
        a = 3;
        b = 5;
        f1(&a, b);
        f2(&a, b);
        cout << a << "," << b << ",";
        cout << a << "," << b;
}

void f1(int* p, int q)
{
        int tmp;
        tmp = *p;
        *p = q;
        q = tmp;
}

void f2(int* p, int q)
{
        int tmp;
        tmp = *p;
        *p = q;
        q = tmp;
}
```

```
int fun2(char* a, char* b)
{
        for (; *a == *b; a++, b++)
                if (*a == '\0')
                        return 0;
        return *a - *b;
}


int main() {
        char a[10] = "date", b[10] = "data";
        cout << fun2(a, b) << endl;
}
```

```
void main()
{
        void* vp;
        char ch = 'g', * cp = "goofy";
        int j = 20;
        vp = &ch;
        cout << *(char*)vp;
        vp = &j;
        cout << *(int*)vp;
        vp = cp;
        cout << (char*)vp + 3 << endl;

}
```

```
int main()
{
	char* ptr;
	char myString[] = "programing I";
	ptr = myString;
	ptr += 5;
	cout << ptr;
}
```

```
int main()
{
	int x = 20;
	int& y = x;
	int* p = &x;
	x = x + 20;
	y = y + 50;
	cout << *p << " " << y;
}
```

```
int main() {

	int data = 10;
	int const* what;
	what = &data;
	cout << what << "\t"
		<< *what << "\\"
		<< &what;
	return 0;
}
```

```
int main()
{
	int array[] = { 1,2,3,4,5 };
	int* p = array;
	cout << (p + (10 - 5) / 2 == array + 1);
	return 0;
}
```

```
int g_One = 1;

void func(int* pInt) {
	pInt = &g_One;
}

void func2(int*& rpInt) {
	rpInt = &g_One;
}

int main() {
	int nvar = 2;
	int* pvar = &nvar;
	func(pvar);
	cout << *pvar << endl;
	func2(pvar);
	cout << *pvar << endl;
	return 0;
}
```

```
int main(){
        char* s[4] = { "black", "white",
                "yellow", "violet" };

        cout << (*(s + 1) + 2) << endl;
        cout << *(*(s + 2) + 3);
        return 0;
}
```

```
void f(int* p, int* q, int* k)
{
        p = q;
        f = p;
        q = f;
        *p = 2, * q = *f + 3; *f = *f + 1;
}
int i = 0, j = 1, f = 6;
int main()
{
        f(&i, &j, &f);
        cout << i << f << j;
        return 0;
}
```

```
void fun(int* p, int* s) {
        s = p;
        *s = 10;
        int x = 5;
        s = &x;
        return;
}
int main() {
        int x = 5;
        int* p = &x;
        int* s;
        s = &x;
        fun(p, s);
        cout << "x = " << x << " *p=" <<
                *p << " *s=" << *s;
        return 0;
}
```

```
const int s = 3;
int* listMystery(int list[][::s]) {
        int i = 1, k = 0;
        int* n = new int[::s];
        for (int i = 0; i < ::s; ++i)
                n[i] = 0;
        while (i < ::s)
        {
                int j = ::s - 1;
                while (j >= i)
                {
                        n[k++] = list[j][i]
                                * list[i][j];
                        j = j - 1;
                }
                i = i + 1;
        }
        return n;
}
void displayMystery(int* arr) {
        cout << "[ ";
        for (int i = 0; i & lt; ::s; ++i)
                cout << arr[i] << (i != (::s - 1)
                        ? " , " : "  ");
        cout << " ] " << endl;
}

int main() {
        int L[][::s] = { {8, 9, 4}, {2, 3, 4},
                {7, 6, 1} };
        int* ptr = listMystery(L);
        displayMystery(ptr);
        delete[] ptr;
        return 0;
}
```

```
void function(char** ptr)
{
        char* ptr1;
        ptr1 = (ptr += sizeof(int))[-2];
        cout << ptr1 << endl;
}

int main()
{
        char* arr[] = { "ant", "bat", "cat",
                "dog", "egg", "fly" };
        function(arr);
        return 0;
}
```

```
int main() {

        int number1 = 88, number2 = 22;
        int* pNumber1 = &number1;
        *pNumber1 = 99;

        cout << *pNumber1 << endl;
        cout << &number1 << endl;
        cout << pNumber1 << endl;
        cout << &pNumber1 << endl;

        pNumber1 = &number2;
        int& refNumber1 = number1;
        refNumber1 = 11;

        cout << refNumber1 << endl;
        cout << &number1 << endl;
        cout << &refNumber1 << endl;

        refNumber1 = number2;
        number2++;

        cout << refNumber1 << endl;
        cout << number1 << endl;
        cout << number2 << endl;
        return 0;
}
```

```
int f(int x, int* py, int** ppz)
```

```
{
        int y, z;
        **ppz += 1;
        z = **ppz;
        *py += 2;
        y = *py;
        x += 3;
        return x + y + z;
}

int main()
{
        int c, * b, ** a;
        c = 4;
        b = &c;
        a = &b;
        cout << f(c, b, a);
        return 0;
}
```

```
int main()
{
  const int* p;
  const int a = 2;
  p = &a;
  *p = 7;
  cout << *p;
}
```

```
int main()
{
  const int a = 2;
  const int* p = &a;
  int b = 3;
  p = &b;
  cout << *p;
}
```

```
int main()
{
        int a[3] = { 1, 2, 3 };
        int* const p = a;

        cout << *(p++);
}
```

```
int main()
{
        int A[2][3] = { {1, 2, 3}, {4, 5, 6} };
        int* p1, * p2;

        int B[3] = { 7, 9, 0 };
        p1 = &B;

        p2 = A;

        cout << *p2;
}
```

```
int main()
{
        void* vp;

        int a = 6;
        float b = 6.9;

        vp = &a;
        cout << *vp;

        vp = &b;
        cout << *vp;
}
```

```
int main()
{
        void* vp;

        int a = 69;

        vp = &a;
        cout << (char*)vp << endl;
        cout << (int*)vp << endl;
        cout << (float*)vp << endl;

}
```

```
int main()
{
        void* vp;
        float b = 6.9;

        vp = &b;
        cout << &b << endl;
        cout << &vp << endl;
        cout << (float*)vp << endl;

        cout << (float**)vp << endl;
        cout << (float***)vp << endl;
        cout << (float******)vp << endl;
}
```

```
int main()
{
        void* vp;

        int a = 69;

        vp = &a;
        cout << &vp << endl;
        cout << &a << endl;

        cout << (void*)vp << endl;
        cout << *(void*)vp << endl;

}
```

```
int main()
{
        void* vp;
```

int a = 69;
 	
    vp = &a;
    cout << &a << endl;
    cout << &vp << endl;
    cout << *(int*)vp << endl;
    cout << *(int*)*&vp << endl;
    cout << (int*)&vp << endl;
 	
}	

int main()
{
    void* vp;
 	
    int a = 69;
 	
    vp = &a;
    cout << &a << endl;
    cout << &vp << endl;
    cout << *(char*)vp << endl;
    cout << (char*)vp << endl;
    cout << (char*&)vp << endl;
    cout << (char*&&)vp << endl;
    cout << *&(char*&)vp << endl;
    cout << (char**)vp << endl;
    cout << (void *)(char*)vp << endl;
    cout << (void *)(char***)vp << endl;
    cout << (char***)&vp << endl;
}	

int main()
{
    void* vp;
    void** vvp = &vp;
    int a = 69;
 	
    vp = &a;
    cout << &a << endl;
    cout << &vp << endl;
    cout << &vvp << endl;
    cout << *vvp << endl;
 	
    cout << (char *)*vvp << endl;
    cout << (void*)(vvp) << endl;
    cout << (**vvp) << endl;
    cout << (char**)(*vvp) << endl;
    cout << &(*vvp) << endl;
 	
    cout << *((char*)*vvp) << endl;
    cout << (void*)(*vvp) << endl;
    cout << (void*)(char*)(vvp) << endl;
    cout << (void*)(void*)(char**)(vvp) << endl;
    cout << (char**)(vvp) << endl;
 	
}	

int main()
{
    void* vp;
    void** vvp;	

```
        int a = 69;
        int* ip = &a;
        vvp = &ip;


        vp = &a;
        cout << &a << endl;
        cout << &vp << endl;
        cout << &vvp << endl;
        cout << *vvp << endl;
        cout << *ip << endl;

}
```

```
int main()
{
        void* vp;
        int a = 69;
        int* ip = &a;
        void* & vvp = vp;


        vp = &a;
        cout << &a << endl;
        cout << &vp << endl;
        cout << &vvp << endl;
        cout << *(char *)vvp << endl;
        cout << (void *)ip << endl;
        cout << (void*)&ip << endl;

        cout << *(int *)vvp + (int **)vp << endl;
        cout << *(int*)vvp + *(int*)vp << endl;

}
```

```
int a = 5;
int b = 6;
int* p = &a;


int* ABC() {
        return &b;
}

int* DEF(int* p) {
        return p;
}

int& DEF() {
        return *p;
}

int& GHI() {
        return a;
}

int main()
{
        int a = 4;
        int* p;
        cout << *(ABC()) << endl;

        p = DEF(&::a);
        cout << *p << endl;

        DEF() = 1;
        cout << ::a << endl;

        a = GHI();
        cout << a << endl;
}
```

```
int main() {

        char pf[] = "PF is an interesting course";
        char* ptr = pf;

        cout << "1 " << ptr[3] << *ptr << endl;
        cout << "2 " << ++ptr << endl;
        cout << "3 " << ++(*pf) << endl;
        cout << "4 " << pf + 5 << endl;
        cout << "5 " << (ptr + 5)[-3] << endl;

}
```

```
int& mystery(int*& p)
{
        static int s = 3;

        if (p)
        {
                cout << *p + 2<< endl;
                delete[] p;
                p = nullptr;
        }

        p = new int[s] {s + s, s + 3, s + 2};

        return s;
}
int& magic(int* p)
{
        if (p)
        {
                cout << *p + 2 << endl;
                delete[] p;
                p = nullptr;
        }

        static int s = 3;
        p = new int[s] {s + s};

        return s;
}
int main()
{
        int* ptr = nullptr;

        mystery(ptr) = 5;
        cout << *ptr << endl;

        mystery(ptr)++;
        cout << *ptr << endl;

        magic(ptr) = 2;
        cout << *ptr << endl;

        if (!ptr)
                cout << "Ok that's All" << endl;
}
```

```
void mystery(int** p)
{
        *p += 2;
        cout << (*p)[1] << endl;
        cout << p[0][-3] << endl;

        p = new int* [3] {*p, *p - 2, *p - 3};

        cout << *(*p + 2) << endl;
        (*p + 3)[-1] = 20;

        delete[] p;

}

void magic(int ptr[][3])
{
        ptr += 1;
        cout << **ptr << endl;
        **ptr += 3;

        cout << ptr[-1][1] << endl;
}


int main()
{
        int arr[3][3] = { 1, 2, 3, 4,
                5, 6, 7, 8, 9 };

        magic(arr + 1);

        int* p = &arr[-1][6];
        mystery(&p);

        p += 2;
        for (int i = 0; i < 3; i++)
                cout << *(p - i) << endl;

}
```

```
int main()
{
        int s = 3, b = 4, t = 7;
        int* ptr = &s;
        int*& rptr = ptr;
        cout << *ptr << endl;
        cout << *rptr << endl;

        ptr = &b;
        cout << *ptr << endl;
        cout << *rptr << endl;

        int* nptr = &t;
        rptr = nptr;

        cout << *nptr << endl;
        cout << *rptr << endl;
        cout << *ptr << endl;

}
```

```
int main()
{
        int s = 3, b = 4, t = 7, & q = t;
        int* ptr = &s;
        int** rptr = &ptr;

        cout << *ptr << endl;
        cout << **rptr << endl;

        ptr = &b;
        cout << *ptr << endl;
        cout << **rptr << endl;

        q = b + 2;
        *rptr = &t;
        cout << **rptr << endl;
        cout << *ptr << endl;

        int* nptr = &s;
        rptr = &nptr;

        cout << *nptr << endl;
        cout << **rptr << endl;
        cout << *ptr << endl;

}
```

```
void find(int, int&, int&, int = 4);
int main() {
        int one = 1, two = 2, three = 3;
        find(one, two, three);

        cout << one << "," << two <<
                "," << three << endl;

        return 0;
}

void find(int a, int& b, int& c, int d) {
        if (d < 1)
                return;
        cout << a << "," << b << "," << c << endl;
        c = a + 2 * b;
        int temp = b;
```

```
        b = a;
        a = 2 * temp;
        d % 2 ? find(b, a, c, d - 1)
                : find(c, b, a, d - 1);
}
```
```
int s = 3;

int func(int a) {

        if (a == 0)
                return func(s--);
        if (a < 0)
        {
                s -= 1;
                cout << s << " , " << a << endl;
                return s + a;
        }

        return func(func(a - 1) - 1);
}

int main() {
        func(2);
}
```
```
void doMagic(int arr[], int n)
{
   if (n <= 1)
      return;

   doMagic(arr, n - 1);

   int last = arr[n - 1];
   int j = n - 2;

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

int main()
{
   int arr[] = { 12, 11, 13, 5, 6, 18};
   int n = sizeof(arr) / sizeof(arr[0]);

   doMagic(arr, n);
   for (int i = 0; i < n; i++)
      cout << arr[i] << " ";
}
```
```
int fun(int a, int b)
{
   if (b == 0)
      return 0;
   if (b % 2 == 0)
      return fun(a + a, b / 2);
```

```
    return fun(a + a, b / 2) + a;
}

int main()
{
    cout << fun(4, 3);
    return 0;
}
```

```
int fun(int n)
{
    if (n > 100)
        return n - 10;
    return fun(fun(n + 11));
}

int main()
{
    cout << " " << fun(99) << " ";
    return 0;
}
```

```
void abc(const char *s)
{
    if (s[0] == '\0')
        return;

    abc(s + 1);
    abc(s + 1);
    cout << s[0];
}

int main()
{
    abc("aneeq");
    return 0;
}
```

```
int fun(int count)
{
    cout << count << endl;
    if (count < 3)
    {
        fun(fun(fun(++count)));
    }
    return count;
}

int main()
{
    fun(1);
    return 0;
}
```

```
int fun(int x, int y)
{
    if (y == 0)
        return 0;

    return (x + fun(x, y - 1));
```

```
}

int main()
{
    cout << fun(2, 3);
    return 0;
}
```
```
int foo(int n, int r) {

    if (n > 0)
        return (n % r + foo(n / r, r));
    else
        return 0;
}

int main()
{
    cout << foo(345, 10);
    return 0;
}
```
```
void crazy(int n, int a, int b)
{
    if (n <= 0)
        return;
    crazy(n - 1, a, b + n);
    cout << n << " " << a << " " << b << endl;
    crazy(n - 1, b, a + n);
}

int main()
{
    crazy(3, 4, 5);
    return 0;
}
```
```
int okay(int n, int m, int PD[4][4])
{

    if (n == 1 || m == 1)
        return PD[n][m] = 1;

    if (PD[n][m] == 0)
    {
        PD[n][m] = okay(n - 1, m, PD)
            + okay(n, m - 1, PD);
    }

    return PD[n][m];
}

int main()
{
    int PD[4][4] = { 0 };

    cout << okay(3, 3, PD);
```

```
    return 0;
}
```

```
void swapIt(char& a, char& b)
{
    char c = a;
    a = b;
    b = c;
}

void foo(char* a, int l, int size)
{
    if (l == size)
        cout << a << endl;
    else {
        for (int i = l; i <= size; i++) {
            swapIt(a[l], a[i]);
            foo(a, l + 1, size);
            swapIt(a[l], a[i]);
        }
    }
}

int main()
{
    char str[] = "ABC";
    foo(str, 0, 2);
}
```

```
int fun(int a, int b)
{
    return a > b ? a : b;
}

int foo(int A[], int n)
{
    if (n == 1)
        return A[0];

    return fun(A[n - 1], foo(A, n - 1));
}

int main()
{
    int A[] = { 1, 4, 45, 6, -50, 10, 2 };
    int n = sizeof(A) / sizeof(A[0]);

    cout << foo(A, n);
    return 0;
}
```

```
void fun(char& a, char& b)
{
    char c = a;
    a = b;
    b = c;
}
```

```
void foo(char * str, int size, int i = 0 )
{
    if (i == size / 2)
        return;

    fun(str[i], str[size - i - 1]);
    foo(str, size, i + 1);
}

int main()
{
    char str[] = "maxe ruoy";
    int size = sizeof(str) / sizeof(str[0]);

    foo(str, size - 1);
    cout << str;

}
```

```
bool foo(int n, int i = 2)
{
    if (n <= 2)
        return (n == 2) ? true : false;
    if (n % i == 0)
        return false;
    if (i * i > n)
        return true;

    return foo(n, i + 1);
}

int main()
{
    int n = 35;
    if (foo(n))
        cout << "Yes";
    else
        cout << "No";

    return 0;
}
```

```
int find(int n)
{
    if (n == 0)
        return 0;
    else
        return (n % 2 + 10 * find(n / 2));
}

int main()
{
    int n = 11;
    cout << find(n);
    return 0;
}
```

```
int print_row(int ct, int num)
{
        if (num == 0)
                return ct;
        cout << ct << "\t";
        print_row(ct + 1, num - 1);
}

void pattern(int n, int count, int num)
{
        if (n == 0)
                return;
        count = print_row(count, num);
        cout << endl;
        pattern(n - 1, count, num + 1);
}

int main()
{
        int n = 5;
        pattern(n, 1, 1);
        return 0;
}
```

```
int print_row(int ct, int num)
{
        if (num == 0)
                return ct;
        cout << ct << "\t";
        print_row(ct + 1, num - 1);
}

void pattern(int n, int count, int num)
{
        if (n == 0)
                return;
        count = print_row(n, num);
        cout << endl;
        pattern(n - 1, count, num + 1);
}

int main()
{
        int n = 5;
        pattern(n, 1, 1);
        return 0;
}
```