1. (40 points) What is the output of this program?

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

struct C
{
    int m = 9;
    static int s;
    C(C &c) { m *= c.m; }
    C(int m) { this->m = m - s; }
    C() { m++; }
    C(int m, int n) { m += n; }
};
int C::s = 2;

int main()
{
    // ------------------------ I
    C w(1, 2), x(3), y, z(w);
    w.s++;
    cout << w.m << endl; // (1)
    cout << x.m << endl; // (2)
    cout << y.m << endl; // (3)
    cout << z.m << endl; // (4)
    cout << x.s << endl; // (5)

    // ------------------------ II
    double a = 42, *b = &a, &c = a, d = a;
    cout << *b << endl; // (6)
    cout << a++ << endl; // (7)
    cout << c << endl; // (8)
    cout << d << endl; // (9)
    cout << (&c == b) << endl; // (10)

    // ------------------------ III
    int m[3][3] = {
        { 19, 22, 80 },
        { 37, 17, 68 },
        { 55, 12, 41 }
    };
    int (*p)[3] = m, *q = m[1];
    cout << m[2][1] << endl; // (11)
    cout << **m << endl; // (12)
    cout << *((p + 1) + 2) << endl; // (13)
    cout << *(q + 1) + 1 << endl; // (14)
    cout << 0[q] << endl; // (15)

    // ------------------------ IV
    const char *str1 = "abcde";
    char str2[] = "abcde";
    cout << strlen(str2) << endl; // (16)
}````
2. (10 points) Write a function `sum_str` that accepts a C-String as an arguments and returns the sum of all the single-digit numbers in the string. For example, if the user enters 2514, the function should return 12, since \(2 + 5 + 1 + 4 = 12\).

3. (15 points) Consider the following function `reverse`, that attempts to reverse an array in place (i.e. without the use of additional storage). It does it by interchanging the first and last elements, then the second and second from last etc. All of the interchanges are done by calling function `interchange`. Here are the two functions and a main program:

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

void interchange(int *xptr, int *yptr)
{
    int temp;
    temp = *xptr;
    *xptr = *yptr;
    *yptr = temp;
}

void reverse(int *ptr, int first, int last)
{
    int *endptr;
    endptr = ptr + last;
    for ( ptr += first; ptr <= endptr; ptr++, endptr-- )
    {
        interchange(ptr, endptr);
    }
}

int main()
{
    int a[3] = { 1, 2, 3 };
    reverse(a, 0, 2);
    for ( int i = 0; i < 3; i++ )
        cout << a[i] << " , ";
}
```

(a) Complete the `interchange` function.
(b) When I compile the program I get a compiler error message complaining of a type mismatch between the definition of `interchange` and its invocation. What exactly is the compiler complaining about?
(c) What very simple change will fix it (so that the program will print 3, 2, 1, ) and why?
4. (15 points) The *Lo Shu Magic Square* is a grid with three rows and three columns that has the following properties:

- The grid contains the numbers 1 through 9 exactly.
- The sum of each row, each column, and each diagonal all add up to the same number. This is shown in Figure 1:

```
4 9 2
3 5 7
8 1 6
```

Figure 1: Lo Shu Magic Square

Write a Boolean function `isMagicSquare` that accepts a two-dimensional $3 \times 3$ array as an argument and returns `true` if it determines it is a Lo Shu Magic Square and `false` if it is not.

5. (20 points) Write a `Circle` class that has the following member variables:

- radius: a private double.
- PI: a static constant double with the value 3.14.

The class should have the following member functions:

- Default Constructor: A default constructor that sets radius to 0.0.
- Another Constructor: Accepts the radius of the circle as an argument.
- `setR`: A mutator function for the radius variable. radius should be greater than 0.
- `getR`: An accessor function for the radius variable.
- `getS`: Returns the area of the circle, which is calculated as $S = \pi r^2$.
- `getC`: Returns the circumference of the circle, which is calculated as $C = 2\pi r$.
- A Destructor: Print `Circle r3.0 is deleted`. The number is the radius of the Circle.