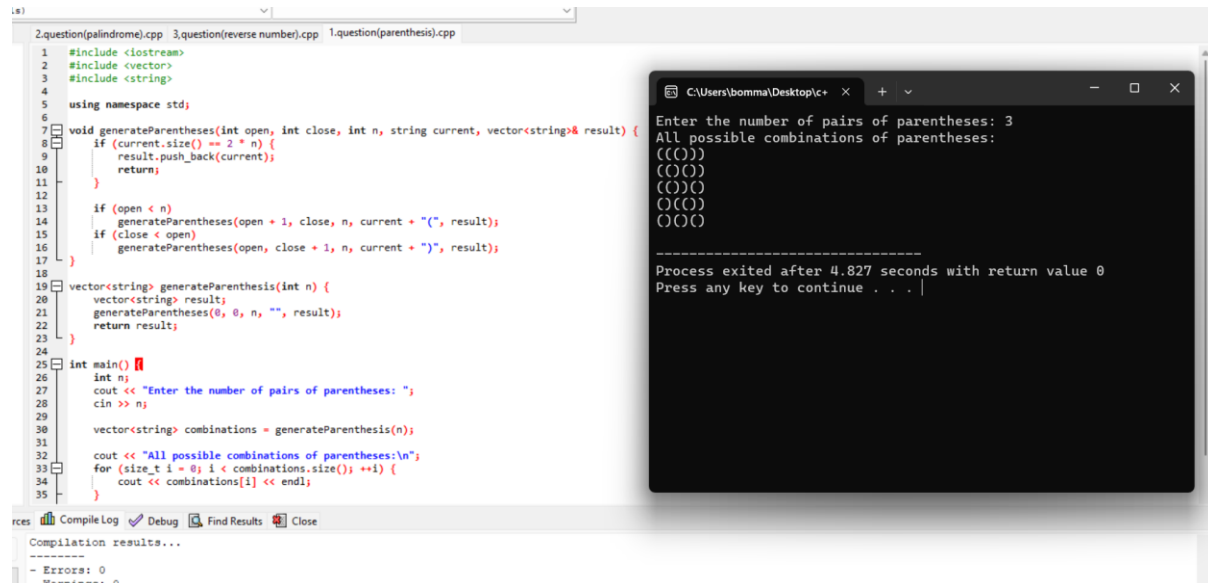


MODEI EXAM OUTPUTS:

1.paranthesis

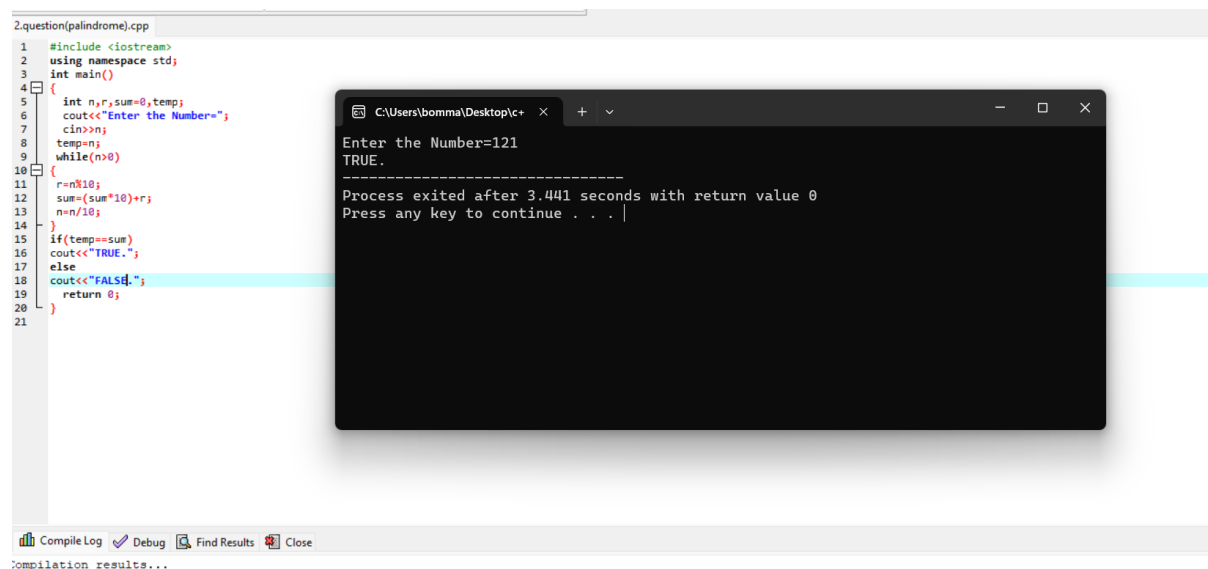


The screenshot shows a C++ IDE with a file named `1.question(parenthesis).cpp`. The code defines a recursive function `generateParentheses` that builds all possible valid parentheses strings for a given number of pairs `n`. The `main` function prompts the user for the number of pairs, calls the function, and prints the results.

```
1 #include <iostream>
2 #include <vector>
3 #include <string>
4
5 using namespace std;
6
7 void generateParentheses(int open, int close, int n, string current, vector<string>& result) {
8     if (current.size() == 2 * n) {
9         result.push_back(current);
10        return;
11    }
12
13    if (open < n)
14        generateParentheses(open + 1, close, n, current + "(", result);
15    if (close < open)
16        generateParentheses(open, close + 1, n, current + ")", result);
17 }
18
19 vector<string> generateParenthesis(int n) {
20     vector<string> result;
21     generateParentheses(0, 0, n, "", result);
22     return result;
23 }
24
25 int main() {
26     int n;
27     cout << "Enter the number of pairs of parentheses: ";
28     cin >> n;
29
30     vector<string> combinations = generateParenthesis(n);
31
32     cout << "All possible combinations of parentheses:\n";
33     for (size_t i = 0; i < combinations.size(); ++i) {
34         cout << combinations[i] << endl;
35     }
36 }
```

The terminal output shows the user entering 3, followed by the 5 possible combinations of 3 pairs of parentheses: `((()))`, `(()())`, `(())()`, `()(())`, and `()()()`. The process exited after 4.827 seconds.

2.palindrome

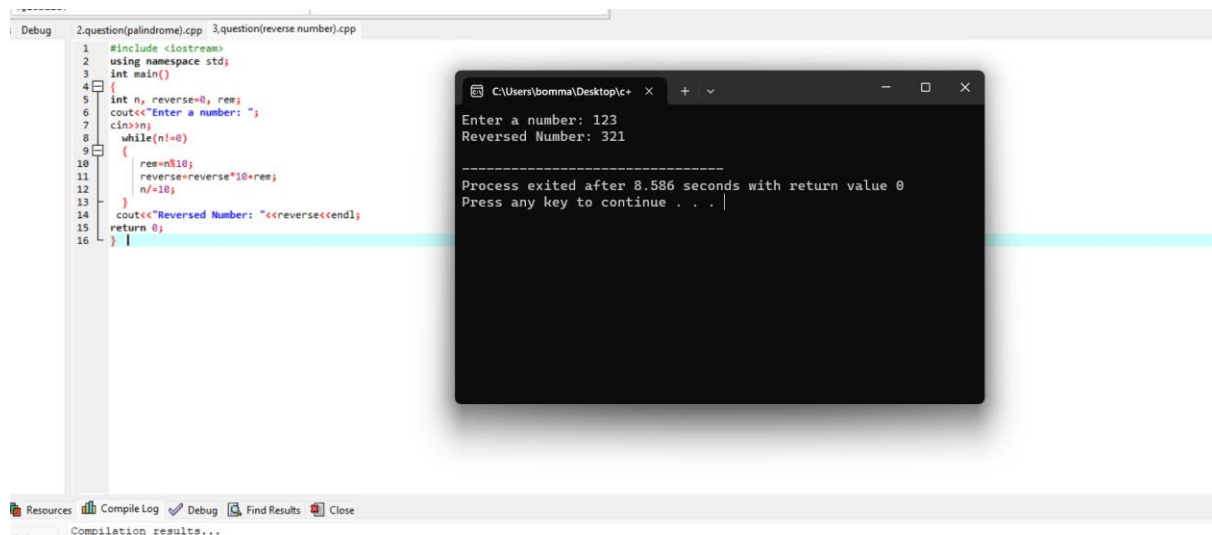


The screenshot shows a C++ IDE with a file named `2.question(palindrome).cpp`. The code implements a function to check if a number is a palindrome by reversing its digits and comparing it to the original number.

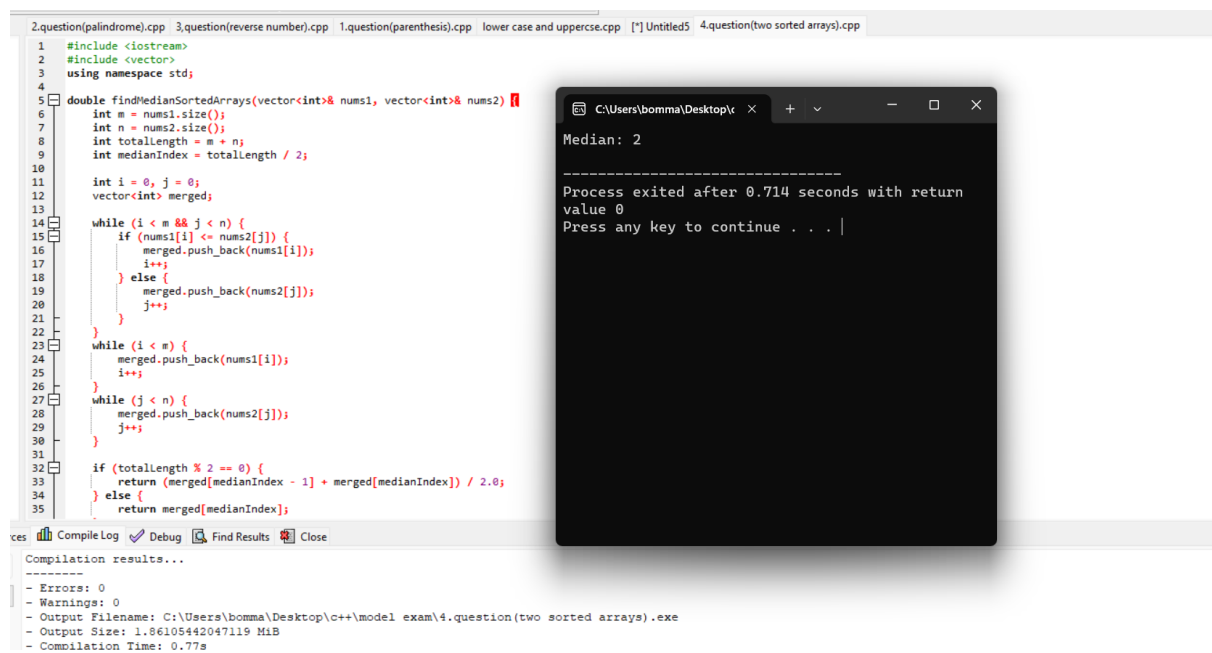
```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int n, r, sum=0, temp;
6     cout << "Enter the Number=";
7     cin >> n;
8     temp=n;
9     while(n>0)
10    {
11        r=n%10;
12        sum=(sum*10)+r;
13        n=n/10;
14    }
15    if(temp==sum)
16        cout << "TRUE.";
17    else
18        cout << "FALSE.";
19    return 0;
20 }
```

The terminal output shows the user entering 121, which is confirmed as a palindrome with the output `TRUE.`. The process exited after 3.441 seconds.

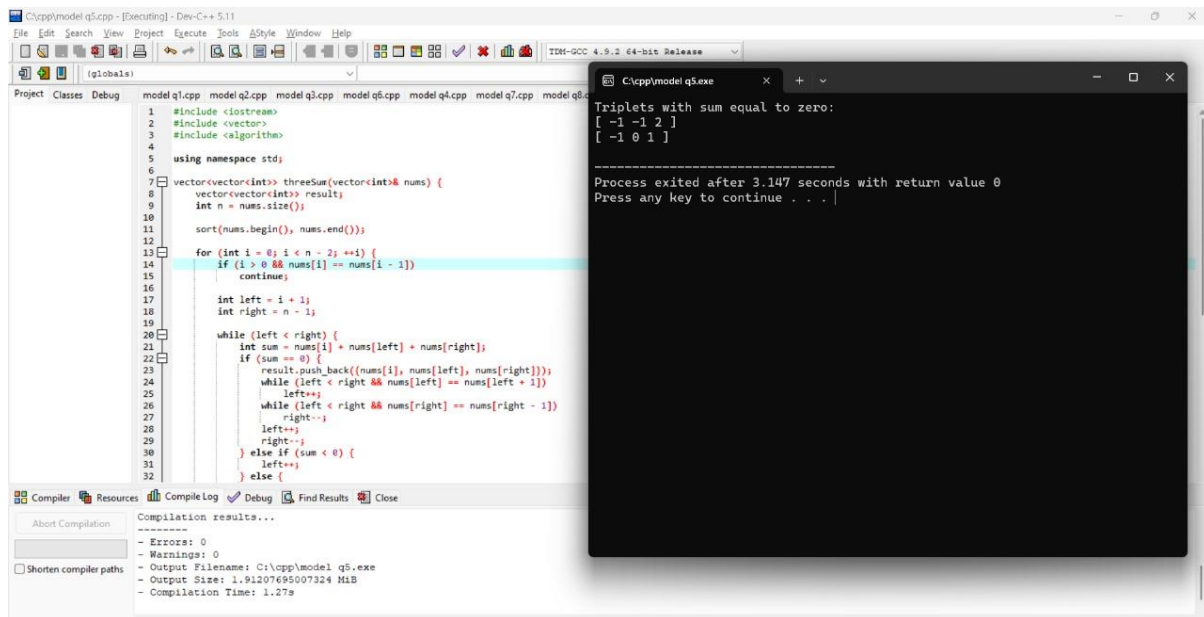
3.Reverse number



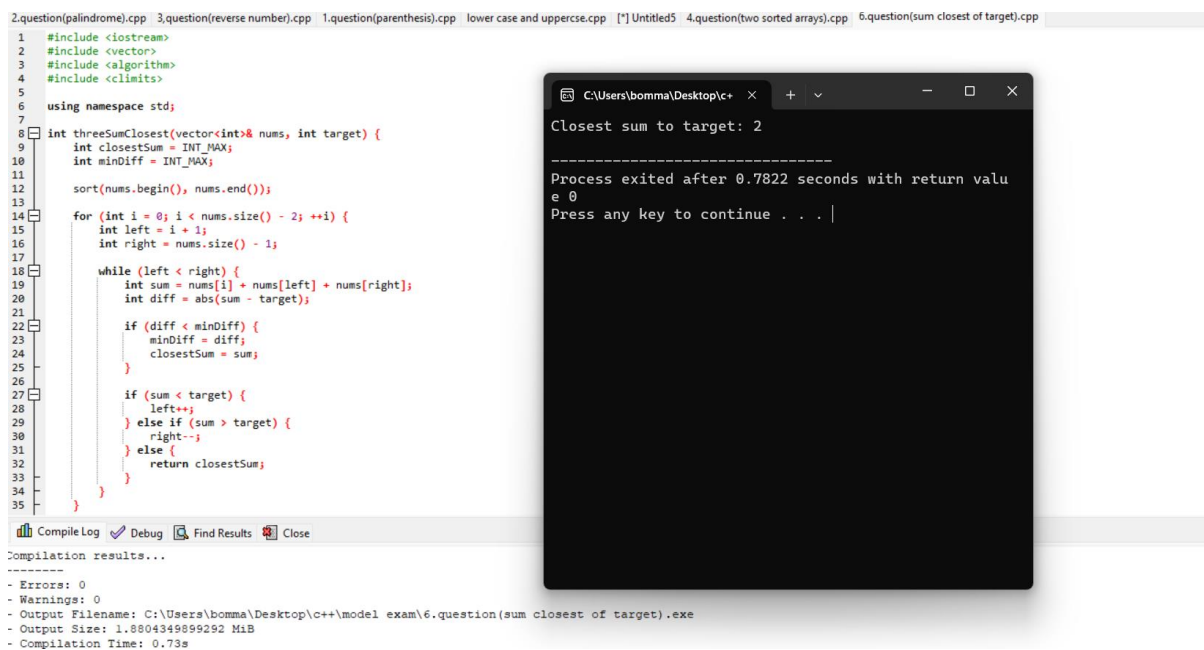
4.Two sorted arrays of num1 and num2:



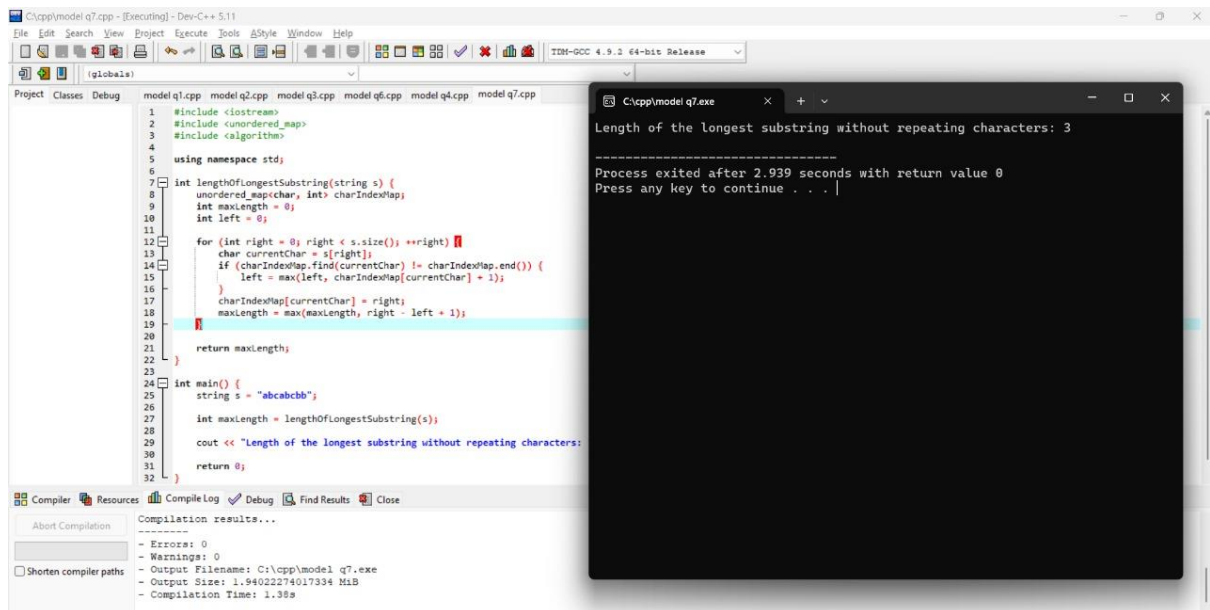
5.triplets:



6.sum is closest to target:



7.length of longest substring:



8.string s and apattern p, implement regular expression:

