Here's an algorithmic representation of the game in a more abstract form. Analyze the algorithm then implement a program in c++.

1. Display "Welcome to the Treasure Quest!"
2. Display "You are on a quest to find the hidden treasure."
3. Initialize an integer variable "choice."
4. Display "You are at a crossroad. Where do you want to go?"
5. Display "1. Go left"
6. Display "2. Go right"
7. Display "3. Go straight"
8. Accept user input into "choice."
9. If "choice" is equal to 1, then:
    1. Display "You chose to go left. You encounter a dangerous river!"
    2. Display "1. Try to swim across"
    3. Display "2. Look for a bridge"
    4. Accept user input into "choice."
    5. If "choice" is equal to 1, then:
        1. Display "You attempt to swim across, but the current is too strong."
        2. Display "You are swept away by the river. Game over!"
    6. Else if "choice" is equal to 2, then:
        1. Display "You find a sturdy bridge and safely cross the river."
        2. Display "Congratulations! You have successfully crossed the river."
        3. Display "You come across a mysterious cave. Do you want to enter?"
        4. Display "1. Enter the cave"
        5. Display "2. Continue on your path"
        6. Accept user input into "choice."
        7. If "choice" is equal to 1, then:
            1. Display "You enter the dark cave. After navigating through some twists and turns,"
            2. Display "you discover a hidden treasure chest!"
            3. Display "Congratulations! You found the hidden treasure. You win!"
        8. Else:
            1. Display "You decide not to enter the cave and continue on your path."
            2. Display "Unfortunately, you miss the hidden treasure. Game over!"
    7. Else:
        1. Display "Invalid choice. Game over!"
10. Else if "choice" is equal to 2, then:
    1. Display "You chose to go right. You stumble upon a group of hostile bandits!"
    2. Display "1. Attempt to fight them"
    3. Display "2. Run away"
    4. Accept user input into "choice."
    5. If "choice" is equal to 1, then:
        1. Display "You bravely fight the bandits, but they overpower you."
        2. Display "You are captured. Game over!"

6. Else if "choice" is equal to 2, then:
    1. Display "You run away from the bandits and escape to safety."
    2. Display "You find yourself in front of an ancient temple."
    3. Display "Do you want to enter the temple?"
    4. Display "1. Enter the temple"
    5. Display "2. Continue on your path"
    6. Accept user input into "choice."
    7. If "choice" is equal to 1, then:
        1. Display "You enter the ancient temple and discover a hidden treasure chest!"
        2. Display "Congratulations! You found the hidden treasure. You win!"
    8. Else:
        1. Display "You decide not to enter the temple and continue on your path."
        2. Display "Unfortunately, you miss the hidden treasure. Game over!"
7. Else:
8. Display "Invalid choice. Game over!"

11. Else if "choice" is equal to 3, then:
    1. Display "You chose to go straight. You encounter a deep forest!"
    2. Display "1. Try to find a path through the forest"
    3. Display "2. Turn back and choose another path"
    4. Accept user input into "choice."
    5. If "choice" is equal to 1, then:
        1. Display "You navigate through the dense forest and get lost."
        2. Display "You cannot find your way out. Game over!"
    6. Else if "choice" is equal to 2, then:
        1. Display "You turn back and choose another path."
        2. Display "You find yourself at the crossroad again."
        3. Display "Choose another path wisely!"
    7. Else:
    8. Display "Invalid choice. Game over!"
12. Else:
    1. Display "Invalid choice. Game over!"
13. Wait for user input (e.g., press a key to exit).
14. End of the game.