```
// Nwakwo Moses
// CSDP 250: Project 2
// Program to convert a number to Binary

#include <iostream>
#include "decimalStack.h"
#include "decimalQueue.h"

void convertToBinary(double, int);//Function Definition for Conversion function

using namespace std;
int main(){
   //Beginning of Program
   cout << "Welcome to Decimal to Binary Program!\n";
   cout << "This program collects any positve number and converts it to decimal.\n";
   cout << "(Note: For Fractions, you will have to indicate the number of decimal places)\n";

   double UserNumber = -1; //Initialize with negative number so while condition runs
successfully.
   //Input for the Number
   while (UserNumber <= 0) {
      cout << "Please input a positive number: ";
      cin >> UserNumber;
      if (UserNumber < 0) {
         cout << "Invalid Input: Only positive numbers are accepted.\n";
      }
   }

   int decimal_place = 0;
   //Input for the decimal Place
   while (decimal_place < 1 || decimal_place > 10) {
      cout << "Please input a positive number for decimal places (1~10): ";
      cin >> decimal_place;
      if (decimal_place < 1 || decimal_place > 10) {
         cout << "Invalid Input: Decimal places should be between 1 and 10.\n";
      }
   }

   convertToBinary(UserNumber, decimal_place);//Call conversion and output function

   cout << "Thank you for using the program!\n";
   return 0;
}
```

```
void convertToBinary(double number, int decimal_places) {
    BinaryStack wholeStack_List;
    BinaryQueue decimalQueue;

    // Separate whole and decimal parts
    int integerPart = number;
    double decimalPart = number - integerPart;

    // Convert whole part to binary using the stack
    if (integerPart == 0) {
        wholeStack_List.push(0);
    }
    else {
        wholeStack_List.pileBinaryDigits(wholeStack_List, integerPart);
    }

    // Convert decimal part to binary using the recursive queue function
    if (decimalPart > 0) {
        decimalQueue.afterDecimalQueue(decimalQueue, decimalPart, decimal_places);
    }

    // Output the binary representation
    cout << "Binary representation of " << number << " is = ";
    // Output whole part
    while (!wholeStack_List.isEmpty()) {
        cout << wholeStack_List.pop();
    }

    // Output fractional part
    if (decimalPart > 0) {
        cout << ".";
        while (!decimalQueue.isEmpty()) {
            cout << decimalQueue.dequeue();
        }
    }
    cout << endl;
}
```