Part 1

Problem 15

File   Edit   Selection   View   Go   Run   Terminal   Help

EXPLORER

∨ OPEN EDITORS
  × lab3part1problem15.cc lab3\finalized
  lab3part1problem19.cc lab3\finalized
  Lab3_prime_numbers.cc lab3\finalized
  sum_digit.cpp lab3\finalized
> 275
> OUTLINE

lab3part1problem15.cc ×   lab3part1problem19.cc   Lab3_prime_numbers.cc   sum_

lab3 > finalized > lab3part1problem15.cc > billGenerator(double, double)

```cpp
1    /*
2     * This C++ program computes and displays a user's hospital charges based on
3     * their status as an inpatient or an outpatient.
4     * CECS 275 - Fall 2021
5     * @author Ethan Hua
6     * @author Rodrigo Becerril Ferreyra
7     * @version 2.3
8     *
9     */
10
11   #include <iostream>
12   #include <string>
13   #include <iomanip>
14
15   using namespace std;
16
17   // Function prototypes
18   void greeter();
19   int patientType();
20   double configuration(int);
21   double billGenerator(int,double,double,double);
22   double billGenerator(double,double);
23
24   int main(){
25
26       greeter();
27
28       int choice = patientType();
29
30       configuration(choice);
31
32       return 0;
33   }
34
35   /**
36    * Text output function that contextualizes the program for the end user.
37    * Defines "inpatient" and "outpatient" to minimize ambiguity.
38    *
39    */
40   void greeter(){
41
42       cout << "------------------------" << endl;
43       cout << "This program calculates a patient's total fees accrued" << endl;
44       cout << "over the course of their hospital attendance." << endl;
45
46       cout << endl;
47
48       cout << "An inpatient is defined as one that required stay at the" << endl;
49       cout << "hospital for at least one full day or overnight." << endl;
```

⊗ 0 ⚠ 0 ⓘ 12   Ln 297, Col 2   Spaces: 4   UTF-8   CRLF   C++   ⚠ 3 Spell   Win32

```cpp
        cout << endl;

        cout << "An outpatient is defined as a patient or individual" << endl;
        cout << "that did not require at least an overnight stay." << endl;

        cout << endl;

        cout << "Was this patient admitted as an inpatient or as an outpatient?"
            << endl;
}

/**
 * Verifies user's patient identity by repeatedly accepting and comparing input
 * until an understood string is passed to the function.
 *
 * @return Patient's identity as inpatient or outpatient as int 0 or 1
 */
int patientType(){

    bool unverified = true;    // loop control variable
    int validatedPatient = 0;  // patient default cause at 0
    string patient = "";       // initialized empty string input

    cout << "Input the admission type now: ";

    // Continuous loop used to verify & prompt user for correct
    // patient identification string.
    do{
        cin >> patient; // user inputs patient type here

        if(patient == "Inpatient" || patient == "inpatient"){
            cout << "You have input: inpatient." << endl;
            cout << "***************************" << endl;
            unverified = false; // disable verification loop
            validatedPatient = 0;
        }
        else if(patient == "Outpatient" || patient == "outpatient"){
            cout << "You have input: outpatient." << endl;
            cout << "***************************" << endl;
            unverified = false; // disable verification loop
            validatedPatient = 1;
        }
        else{ // catch-all for typos, integer input, or other irregularities
            cout<<"Could not identify patient type." << endl;
            cout<<"Please enter either \"Inpatient\" or \"Outpatient\"."
                << endl;
        }
```

File   Edit   Selection   View   Go   Run   Terminal   Help

EXPLORER   ...

OPEN EDITORS
  × lab3part1problem15.cc lab3\finalized
     lab3part1problem19.cc lab3\finalized
     Lab3_prime_numbers.cc lab3\finalized
     sum_digit.cpp lab3\finalized
> 275 lab3\finalized
> OUTLINE

lab3part1problem15.cc ×   lab3part1problem19.cc   Lab3_prime_numbers.cc   sum_

lab3 > finalized > lab3part1problem15.cc > billGenerator(double, double)

```cpp
 99         }while(unverified);
100
101         return validatedPatient;
102     }
103
104     /**
105     * Uses integer input representation of patient type to prompt fee
106     * data collection from user, then prints bill.
107     * Function continually accepts input for a list of parameters based on
108     * patient status until all inputs confirmed valid, and finishes by printing
109     * an itemized list of fees as well as total bill using user's submissions.
110     * @param patientData integer 0 or 1 used to denote inpatient or outpatient
111     * @return sum total of all user's fees
112     */
113     double configuration(int patientData){
114
115         // fee storage variables
116         int duration = 0;               // length of stay (inpatients)
117         double hospitalFee = 0;         // rate per day of stay (inpatients)
118         double dailyCharges = 0;        // product of duration and hospitalFee
119         double medicalFee = 0;          // fees for medication
120         double servicesFee = 0;         // fees for staff service
121         double finalCharges = 0;        // total of all above fees
122
123         // verification loop control variables
124         bool validDays = false;
125         bool validHospitalFee = false;
126         bool validMedicalFee = false;
127         bool validServicesFee = false;
128
129         if(patientData == 0){ // inpatient case
130
131             do{
132                 cout << "Input the number of days admitted: ";
133                 cin >> duration;
134                 if(duration > 0){ // inpatient stays for at least 1 day
135                     validDays = true; // disable verification loop
136                 }
137                 else{
138                     cout << "Not a valid day count." << endl;
139                 } // Continues prompting input until greater than 0
140             }while(validDays == false);
141
142             do{
143                 cout << "Input the daily rate charged (in USD): $";
144                 cin >> hospitalFee;
145                 if(hospitalFee >= 0){ // user pays no money or any value above 0
146                     validHospitalFee = true; // disable verification loop
147                 }
```

⊗ 0 ⚠ 0 ⓘ 12   Ln 297, Col 2   Spaces: 4   UTF-8   CRLF   C++   ⚠ 3 Spell   Win32

File   Edit   Selection   View   Go   Run   Terminal   Help

EXPLORER   ...

OPEN EDITORS
  ✕ lab3part1problem15.cc lab3\finalized
     lab3part1problem19.cc lab3\finalized
     Lab3_prime_numbers.cc lab3\finalized
     sum_digit.cpp lab3\finalized
> 275
> OUTLINE

lab3part1problem15.cc ✕   lab3part1problem19.cc   Lab3_prime_numbers.cc   sum_

lab3 > finalized > lab3part1problem15.cc > billGenerator(double, double)

```cpp
148          else{
149              cout << "Not a valid fee." << endl;
150          } // Continues prompting input until greater than or equal to 0
151      }while(validHospitalFee == false);
152
153      do{
154          cout << "Input the medication charges (in USD): $";
155          cin >> medicalFee;
156          if(medicalFee >= 0){ // user pays no money or any value above 0
157              validMedicalFee = true; // disable verification loop
158          }
159          else{
160              cout << "Not a valid fee." << endl;
161          } // Continues prompting input until greater than or equal to 0
162      }while(validMedicalFee == false);
163
164      do{
165          cout << "Input the services charges (in USD): $";
166          cin >> servicesFee;
167          if(servicesFee >= 0){ // user pays no money or any value above 0
168              validServicesFee = true; // disable verification loop
169          }
170          else{
171              cout << "Not a valid fee." << endl;
172          } // Continues prompting input until greater than or equal to 0
173      }while(validServicesFee == false);
174
175      // calculates total of daily fees for printout
176      dailyCharges = (duration*hospitalFee);
177
178      // overloaded function call --> case 1
179      finalCharges = billGenerator(duration,hospitalFee,medicalFee,servicesFee);
180
181      // format top of output bill
182
183      cout << ("#####################################") << endl;
184      cout << "Services Rendered" << endl;
185      cout << "*****************" << endl;
186
187      if(duration == 1){ // specific formatting for only 1 day
188          cout << "Patient was admitted for " << duration
189          << "day." << endl;
190      }
191      else{ // "days" plural for all other possible numbers
192          cout << "Patient was admitted for " << duration
193          << " days." << endl;
194      }
195
196      cout << "Daily Rate...............: $" << setprecision(2) <<
```

⊗ 0 ⚠ 0 ⓘ 12   Ln 297, Col 2   Spaces: 4   UTF-8   CRLF   C++   ⚠ 3 Spell   Win32
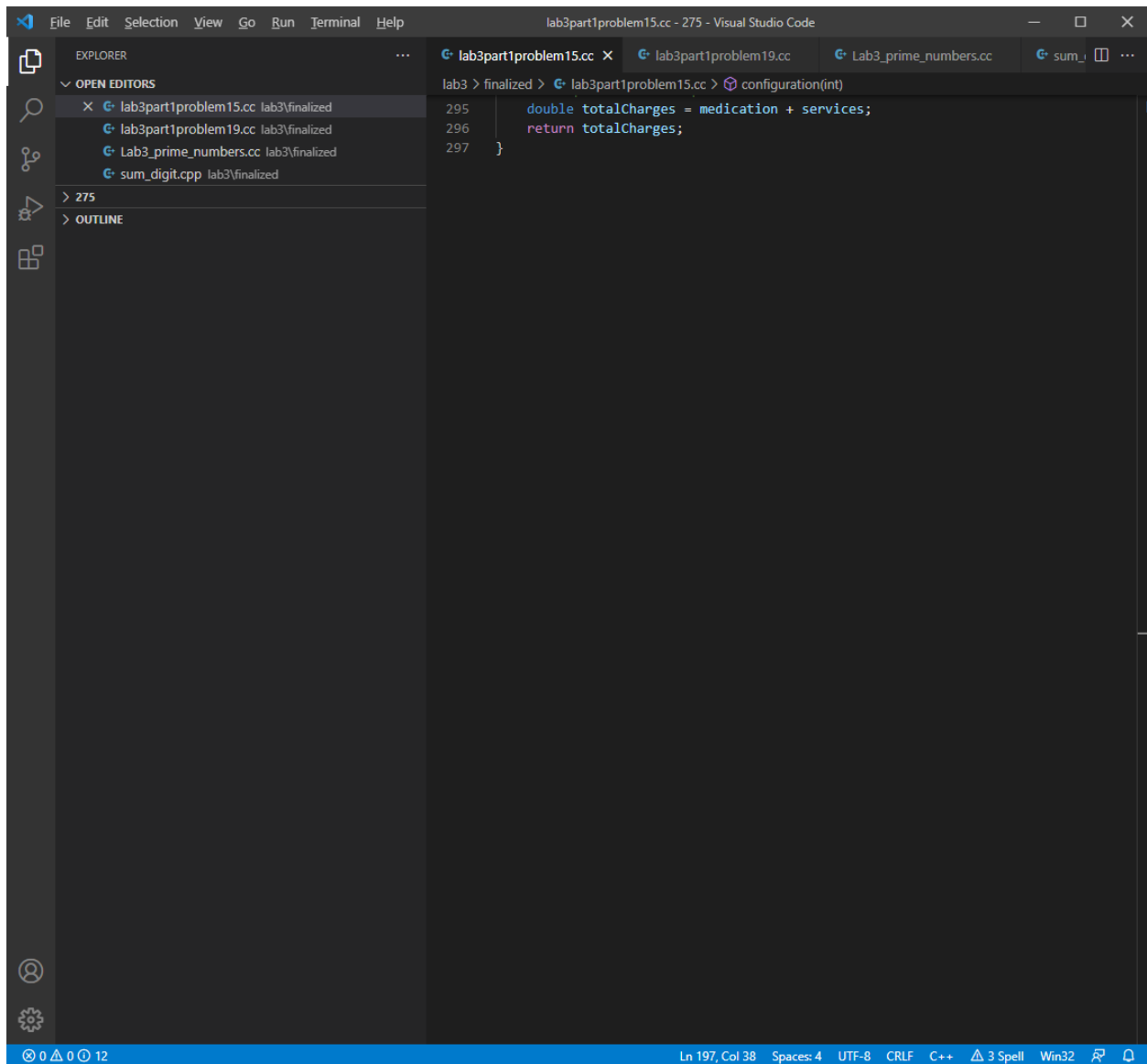
EXPLORER

OPEN EDITORS
- lab3part1problem15.cc lab3\finalized
- lab3part1problem19.cc lab3\finalized
- Lab3_prime_numbers.cc lab3\finalized
- sum_digit.cpp lab3\finalized

> 275
> OUTLINE

lab3part1problem15.cc ×   lab3part1problem19.cc   Lab3_prime_numbers.cc   sum_

lab3 > finalized > lab3part1problem15.cc > configuration(int)

```cpp
197            fixed << hospitalFee << endl;
198            cout << "    Daily Total.........: $" << setprecision(2) <<
199            fixed << dailyCharges << endl;
200            cout << "Medication Fees.........: $" << setprecision(2) <<
201            fixed << medicalFee << endl;
202            cout << "Services Fees...........: $" << setprecision(2) <<
203            fixed << servicesFee << endl;
204
205            cout << endl;
206            cout << "Final charges incurred...: $" << setprecision(2)
207            << fixed << finalCharges << endl;
208
209            cout << ("#######################################") << endl;
210
211            return finalCharges;
212        }
213
214        else if(patientData == 1){ // outpatient case
215
216            do{
217                cout << "Input the medication charges (in USD): $";
218                cin >> medicalFee;
219                if(medicalFee >= 0){ // user pays no money or any value above 0
220                    validMedicalFee = true; // disable verification loop
221                }
222                else{
223                    cout << "Not a valid fee." << endl;
224                } // Continues prompting input until greater than or equal to 0
225            }while(validMedicalFee == false);
226
227            do{
228                cout << "Input the services charges (in USD): $";
229                cin >> servicesFee;
230                if(servicesFee >= 0){ // user pays no money or any value above 0
231                    validServicesFee = true; // disable verification loop
232                }
233                else{
234                    cout << "Not a valid fee." << endl;
235                } // Continues prompting input until greater than or equal to 0
236            }while(validServicesFee == false);
237
238            // overloaded function call --> case 2
239            finalCharges = billGenerator(medicalFee,servicesFee);
240
241            // format top of output bill
242
243            cout << ("#######################################") << endl;
244            cout << "Services Rendered" << endl;
245            cout << "*****************" << endl;
```

File   Edit   Selection   View   Go   Run   Terminal   Help

EXPLORER

OPEN EDITORS
 × lab3part1problem15.cc lab3\finalized
   lab3part1problem19.cc lab3\finalized
   Lab3_prime_numbers.cc lab3\finalized
   sum_digit.cpp lab3\finalized
> 275
> OUTLINE

lab3part1problem15.cc ×   lab3part1problem19.cc   Lab3_prime_numbers.cc   sum_

lab3 > finalized > lab3part1problem15.cc > configuration(int)

```cpp
246
247            cout << "Medication Fees..........: $" << setprecision(2) <<
248            fixed << medicalFee << endl;
249            cout << "Services Fees...........: $" << setprecision(2) <<
250            fixed << servicesFee << endl;
251
252            cout << endl;
253            cout << "Final charges incurred...: $" << setprecision(2)
254            << fixed << finalCharges << endl;;
255
256            cout << ("#####################################") << endl;
257
258            return finalCharges;
259        }
260
261        else{ // catch-all case for unexpected parameter value
262            cout << "Error. Could not generate bill." << endl;
263            finalCharges = 0;
264            return finalCharges;
265        }
266    }
267
268    /**
269     * Calculates sum of fees input by user, assuming they are an inpatient.
270     * Overloaded function case 1 defined by problem 15.
271     * Differentiated by int parameter in this definition.
272     *
273     * @param days duration of stay in hospital
274     * @param rate fee paid per day spent in hospital
275     * @param medication aggregate cost of medications used on patient
276     * @param services fees for hospital equipment upkeep/use
277     */
278    double billGenerator(int days, double rate, double medication, double services){
279        // total daily fee generated by days multiplied by rate
280        // then summed with flat fees
281        double totalCharges = (days * rate) + medication + services;
282        return totalCharges;
283    }
284
285    /**
286     * Calculates sum of fees input by user, assuming they are an outpatient.
287     * Overloaded function case 2 defined by problem 15.
288     * Differentiated by use of only 2 parameters.
289     *
290     * @param medication aggregate cost of medications used on patient
291     * @param services fees for hospital equipment upkeep/use
292     */
293    double billGenerator(double medication, double services){
294        // simple sum of two parameters
```

⊗ 0 ⚠ 0 ⓘ 12   Ln 197, Col 38   Spaces: 4   UTF-8   CRLF   C++   ⚠ 3 Spell   Win32

lab3 > finalized > lab3part1problem15.cc > configuration(int)

```cpp
    double totalCharges = medication + services;
    return totalCharges;
}
```

lab3part1problem15.cc - 275 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

∨ OPEN EDITORS
  ✕ lab3part1problem15.cc lab3\finalized
     lab3part1problem19.cc lab3\finalized
     Lab3_prime_numbers.cc lab3\finalized
     sum_digit.cpp lab3\finalized
∨ 275
∨ OUTLINE

TERMINAL   DEBUG CONSOLE   PROBLEMS 12   OUTPUT   powershell

```
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3\finalized> g++ .\lab3part1problem15.cc
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3\finalized> .\a.exe
------------------------
This program calculates a patient's total fees accrued
over the course of their hospital attendance.

An inpatient is defined as one that required stay at the
hospital for at least one full day or overnight.

An outpatient is defined as a patient or individual
that did not require at least an overnight stay.

Was this patient admitted as an inpatient or as an outpatient?
Input the admission type now: Isndfp
Could not identify patient type.
Please enter either "Inpatient" or "Outpatient".
1242
Could not identify patient type.
Please enter either "Inpatient" or "Outpatient".
Inpatient
You have input: inpatient.
**************************
Input the number of days admitted: -95
Not a valid day count.
Input the number of days admitted: 0
Not a valid day count.
Input the number of days admitted: 13
Input the daily rate charged (in USD): $-79
Not a valid fee.
Input the daily rate charged (in USD): $-2135.21
Not a valid fee.
Input the daily rate charged (in USD): $79.68
Input the medication charges (in USD): $-979
Not a valid fee.
Input the medication charges (in USD): $-1965.28
Not a valid fee.
Input the medication charges (in USD): $1564.87
Input the services charges (in USD): $-918
Not a valid fee.
Input the services charges (in USD): $-2318.15
Not a valid fee.
Input the services charges (in USD): $4015.88
######################################
Services Rendered
*****************
Patient was admitted for 13 days.
Daily Rate...............: $79.68
    Daily Total..........: $1035.84
Medication Fees..........: $1564.87
Services Fees............: $4015.88

Final charges incurred...: $6616.59
######################################
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3\finalized>
```

⊗ 0 ⚠ 0 ⓘ 12   Ln 11, Col 20   Spaces: 4   UTF-8   CRLF   C++   ⚠ 3 Spell   Win32

File Edit Selection View Go Run Terminal Help

EXPLORER

∨ OPEN EDITORS
  ✕ G⁺ lab3part1problem15.cc lab3\finalized
    G⁺ lab3part1problem19.cc lab3\finalized
    G⁺ Lab3_prime_numbers.cc lab3\finalized
    G⁺ sum_digit.cpp lab3\finalized
> 275
> OUTLINE

TERMINAL   DEBUG CONSOLE   PROBLEMS 12   OUTPUT

```
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3\finalized> g++ .\lab3part1problem15.cc
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3\finalized> .\a.exe
------------------------
This program calculates a patient's total fees accrued
over the course of their hospital attendance.

An inpatient is defined as one that required stay at the
hospital for at least one full day or overnight.

An outpatient is defined as a patient or individual
that did not require at least an overnight stay.

Was this patient admitted as an inpatient or as an outpatient?
Input the admission type now: uefhi
Could not identify patient type.
Please enter either "Inpatient" or "Outpatient".
134214
Could not identify patient type.
Please enter either "Inpatient" or "Outpatient".
Outpatient
You have input: outpatient.
***************************
Input the medication charges (in USD): $-4595465
Not a valid fee.
Input the medication charges (in USD): $-849.15
Not a valid fee.
Input the medication charges (in USD): $368.54
Input the services charges (in USD): $-9871
Not a valid fee.
Input the services charges (in USD): $-9845.15
Not a valid fee.
Input the services charges (in USD): $1468.23
#######################################
Services Rendered
*****************
Medication Fees..........: $368.54
Services Fees............: $1468.23

Final charges incurred...: $1836.77
#######################################
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3\finalized>
```

Ln 11, Col 20   Spaces: 4   UTF-8   CRLF   C++   ⚠ 3 Spell   Win32

```
/*
 * This C++ program computes and displays a user's hospital charges based on
 * their status as an inpatient or an outpatient. This version generates the
 * printout in an external .txt file.
 * CECS 275 - Fall 2021
 * @author Ethan Hua
 * @author Rodrigo Becerril Ferreyra
 * @version 2.4
 *
 */

#include <iostream>
#include <string>
#include <fstream>
#include <iomanip>

using namespace std;

// Function prototypes
void greeter();
int patientType();
double configuration(int);
double billGenerator(int,double,double,double);
double billGenerator(double,double);

int main(){

    greeter();

    int choice = patientType();

    configuration(choice);

    return 0;
}

/**
 * Text output function that contextualizes the program for the end user.
 * Defines "inpatient" and "outpatient" to minimize ambiguity.
 *
 */
void greeter(){

    cout << "-------------------------" << endl;

    cout << "This program calculates a patient's total fees accrued" << endl;
    cout << "over the course of their hospital attendance." << endl;

```

```
49        cout << endl;
50
51        cout << "An inpatient is defined as one that required stay at the" << endl;
52        cout << "hospital for at least one full day or overnight." << endl;
53
54        cout << endl;
55
56        cout << "An outpatient is defined as a patient or individual" << endl;
57        cout << "that did not require at least an overnight stay." << endl;
58
59        cout << endl;
60
61        cout << "Was this patient admitted as an inpatient or as an outpatient?"
62            << endl;
63 }
64
65 /**
66  * Verifies user's patient identity by repeatedly accepting and comparing input
67  * until an understood string is passed to the function.
68  *
69  * @return Patient's identity as inpatient or outpatient as int 0 or 1
70  */
71 int patientType(){
72
73     bool unverified = true;    // loop control variable
74     int validatedPatient = 0;  // patient default cause at 0
75     string patient = "";       // initialized empty string input
76
77     cout << "Input the admission type now." << endl;
78
79     // Continuous loop used to verify & prompt user for correct
80     // patient identification string.
81     do{
82         cin >> patient; // user inputs patient type here
83
84         if(patient == "Inpatient" || patient == "inpatient"){
85             cout << "You have input: inpatient." << endl;
86             cout << "***************************" << endl;
87             unverified = false; // disable verification loop
88             validatedPatient = 0;
89         }
90         else if(patient == "Outpatient" || patient == "outpatient"){
91             cout << "You have input: outpatient." << endl;
92             cout << "***************************" << endl;
93             unverified = false; // disable verification loop
94             validatedPatient = 1;
95         }
96         else{ // catch-all for typos, integer input, or other irregularities
```

```cpp
 97              cout<<"Could not identify patient type." << endl;
 98              cout<<"Please enter either \"Inpatient\" or \"Outpatient\"."
 99                  << endl;
100          }
101
102      }while(unverified);
103
104      return validatedPatient;
105  }
106
107  /**
108   * Uses integer input representation of patient type to prompt fee
109   * data collection from user, then prints bill.
110   * Function continually accepts input for a list of parameters based on
111   * patient status until all inputs confirmed valid, and finishes by printing
112   * an itemized list of fees as well as total bill using user's submissions.
113   * @param patientData integer 0 or 1 used to denote inpatient or outpatient
114   * @return sum total of all user's fees
115   */
116  double configuration(int patientData){
117
118      ofstream outFile; // ready outgoing file storage for Q19
119
120      // fee storage variables
121      int duration = 0;            // length of stay (inpatients)
122      double hospitalFee = 0;      // rate per day of stay (inpatients)
123      double dailyCharges = 0;     // product of duration and hospitalFee
124      double medicalFee = 0;       // fees for medication
125      double servicesFee = 0;      // fees for staff service
126      double finalCharges = 0;     // total of all above fees
127
128      // verification loop control variables
129      bool validDays = false;
130      bool validHospitalFee = false;
131      bool validMedicalFee = false;
132      bool validServicesFee = false;
133
134      outFile.open("bill.txt"); // uses "bill.txt" in directory to store summary
135
136      if(patientData == 0){ // inpatient case
137
138          do{
139              cout << "Input the number of days admitted: ";
140              cin >> duration;
141              if(duration > 0){ // inpatient stays for at least 1 day
142                  validDays = true; // disable verification loop
143              }
144              else{
```
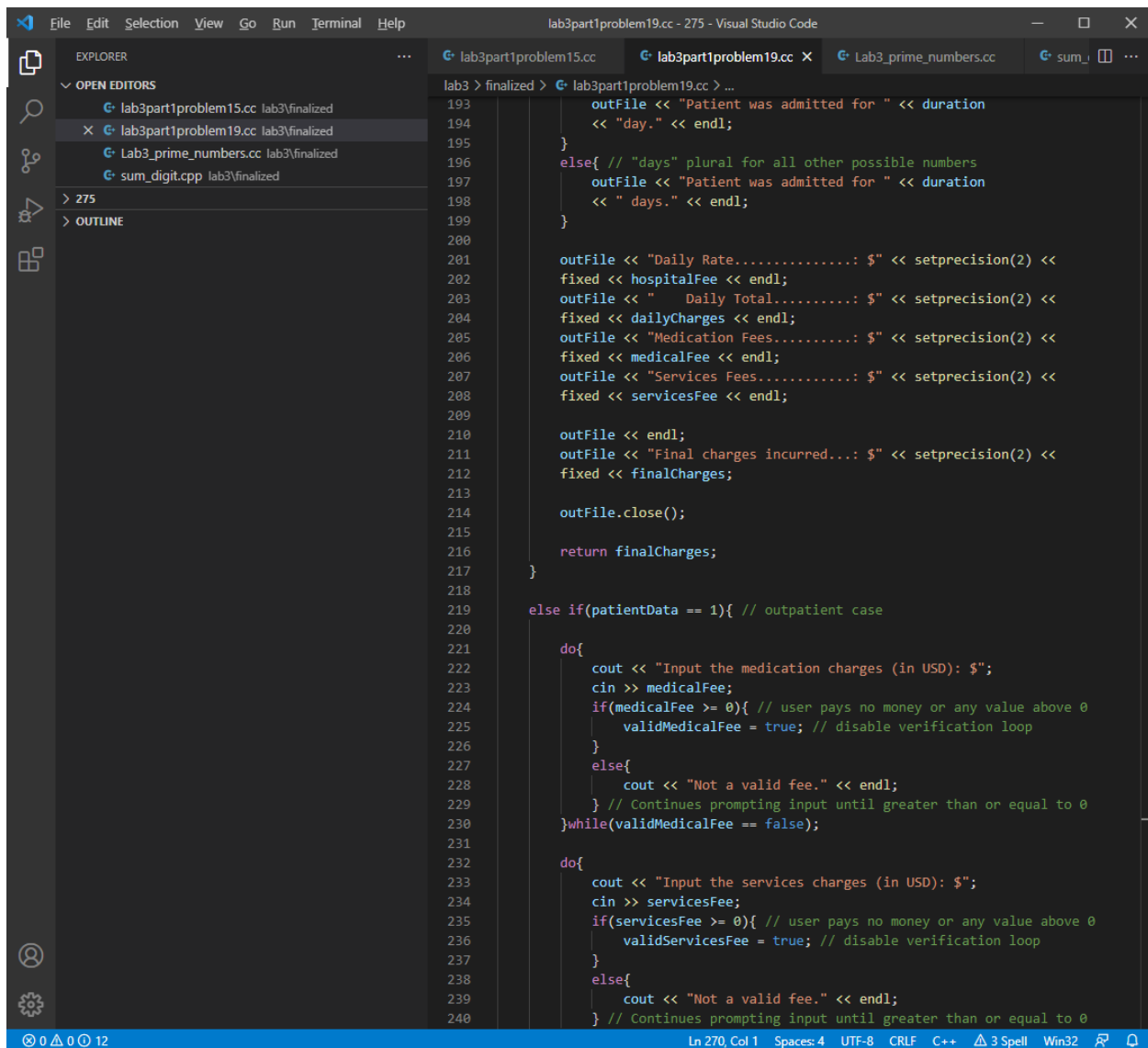
File   Edit   Selection   View   Go   Run   Terminal   Help

EXPLORER

∨ OPEN EDITORS
   lab3part1problem15.cc lab3\finalized
× lab3part1problem19.cc lab3\finalized
   Lab3_prime_numbers.cc lab3\finalized
   sum_digit.cpp lab3\finalized
> 275
> OUTLINE

lab3part1problem15.cc   lab3part1problem19.cc ×   Lab3_prime_numbers.cc   sum_

lab3 > finalized > lab3part1problem19.cc > ...

```cpp
145            cout << "Not a valid day count." << endl;
146        } // Continues prompting input until greater than 0
147    }while(validDays == false);
148
149    do{
150        cout << "Input the daily rate charged (in USD): $";
151        cin >> hospitalFee;
152        if(hospitalFee >= 0){ // user pays no money or any value above 0
153            validHospitalFee = true; // disable verification loop
154        }
155        else{
156            cout << "Not a valid fee." << endl;
157        } // Continues prompting input until greater than or equal to 0
158    }while(validHospitalFee == false);
159
160    do{
161        cout << "Input the medication charges (in USD): $";
162        cin >> medicalFee;
163        if(medicalFee >= 0){ // user pays no money or any value above 0
164            validMedicalFee = true; // disable verification loop
165        }
166        else{
167            cout << "Not a valid fee." << endl;
168        } // Continues prompting input until greater than or equal to 0
169    }while(validMedicalFee == false);
170
171    do{
172        cout << "Input the services charges (in USD): $";
173        cin >> servicesFee;
174        if(servicesFee >= 0){ // user pays no money or any value above 0
175            validServicesFee = true; // disable verification loop
176        }
177        else{
178            cout << "Not a valid fee." << endl;
179        } // Continues prompting input until greater than or equal to 0
180    }while(validServicesFee == false);
181
182    // calculates total of daily fees for printout
183    dailyCharges = (duration*hospitalFee);
184
185    // overloaded function call --> case 1
186    finalCharges = billGenerator(duration,hospitalFee,medicalFee,servicesFee);
187
188    // format top of outgoing saved file
189    outFile << "Services Rendered" << endl;
190    outFile << "*****************" << endl;
191
192    if(duration == 1){ // specific formatting for only 1 day
```

⊗ 0 ⚠ 0 ⓘ 12   Ln 270, Col 1   Spaces: 4   UTF-8   CRLF   C++   △ 3 Spell   Win32

```cpp
                 outFile << "Patient was admitted for " << duration
                     << "day." << endl;
             }
             else{ // "days" plural for all other possible numbers
                 outFile << "Patient was admitted for " << duration
                     << " days." << endl;
             }

             outFile << "Daily Rate...............: $" << setprecision(2) <<
             fixed << hospitalFee << endl;
             outFile << "    Daily Total..........: $" << setprecision(2) <<
             fixed << dailyCharges << endl;
             outFile << "Medication Fees..........: $" << setprecision(2) <<
             fixed << medicalFee << endl;
             outFile << "Services Fees............: $" << setprecision(2) <<
             fixed << servicesFee << endl;

             outFile << endl;
             outFile << "Final charges incurred...: $" << setprecision(2) <<
             fixed << finalCharges;

             outFile.close();

             return finalCharges;
         }

         else if(patientData == 1){ // outpatient case

             do{
                 cout << "Input the medication charges (in USD): $";
                 cin >> medicalFee;
                 if(medicalFee >= 0){ // user pays no money or any value above 0
                     validMedicalFee = true; // disable verification loop
                 }
                 else{
                     cout << "Not a valid fee." << endl;
                 } // Continues prompting input until greater than or equal to 0
             }while(validMedicalFee == false);

             do{
                 cout << "Input the services charges (in USD): $";
                 cin >> servicesFee;
                 if(servicesFee >= 0){ // user pays no money or any value above 0
                     validServicesFee = true; // disable verification loop
                 }
                 else{
                     cout << "Not a valid fee." << endl;
                 } // Continues prompting input until greater than or equal to 0
```

```cpp
241            }while(validServicesFee == false);
242
243            // overloaded function call --> case 2
244            finalCharges = billGenerator(medicalFee,servicesFee);
245
246            // format top of outgoing saved file
247            outFile << "Services Rendered" << endl;
248            outFile << "*****************" << endl;
249
250            outFile << "Medication Fees..........: $" << setprecision(2) <<
251            fixed << medicalFee << endl;
252            outFile << "Services Fees............: $" << setprecision(2) <<
253            fixed << servicesFee << endl;
254
255            outFile << endl;
256            outFile << "Final charges incurred...: $" << setprecision(2)
257            << fixed << finalCharges;
258
259            outFile.close();
260
261            return finalCharges;
262        }
263
264        else{ // catch-all case for unexpected parameter value
265            cout << "Error. Could not generate bill." << endl;
266            finalCharges = 0;
267            return finalCharges;
268        }
269    }
270
271    /**
272    * Calculates sum of fees input by user, assuming they are an inpatient.
273    * Overloaded function case 1 defined by problem 15.
274    * Differentiated by int parameter in this definition.
275    *
276    * @param days duration of stay in hospital
277    * @param rate fee paid per day spent in hospital
278    * @param medication aggregate cost of medications used on patient
279    * @param services fees for hospital equipment upkeep/use
280    */
281    double billGenerator(int days, double rate, double medication, double services){
282        // total daily fee generated by days multiplied by rate
283        // then summed with flat fees
284        double totalCharges = (days * rate) + medication + services;
285        return totalCharges;
286    }
287
288    /**
```
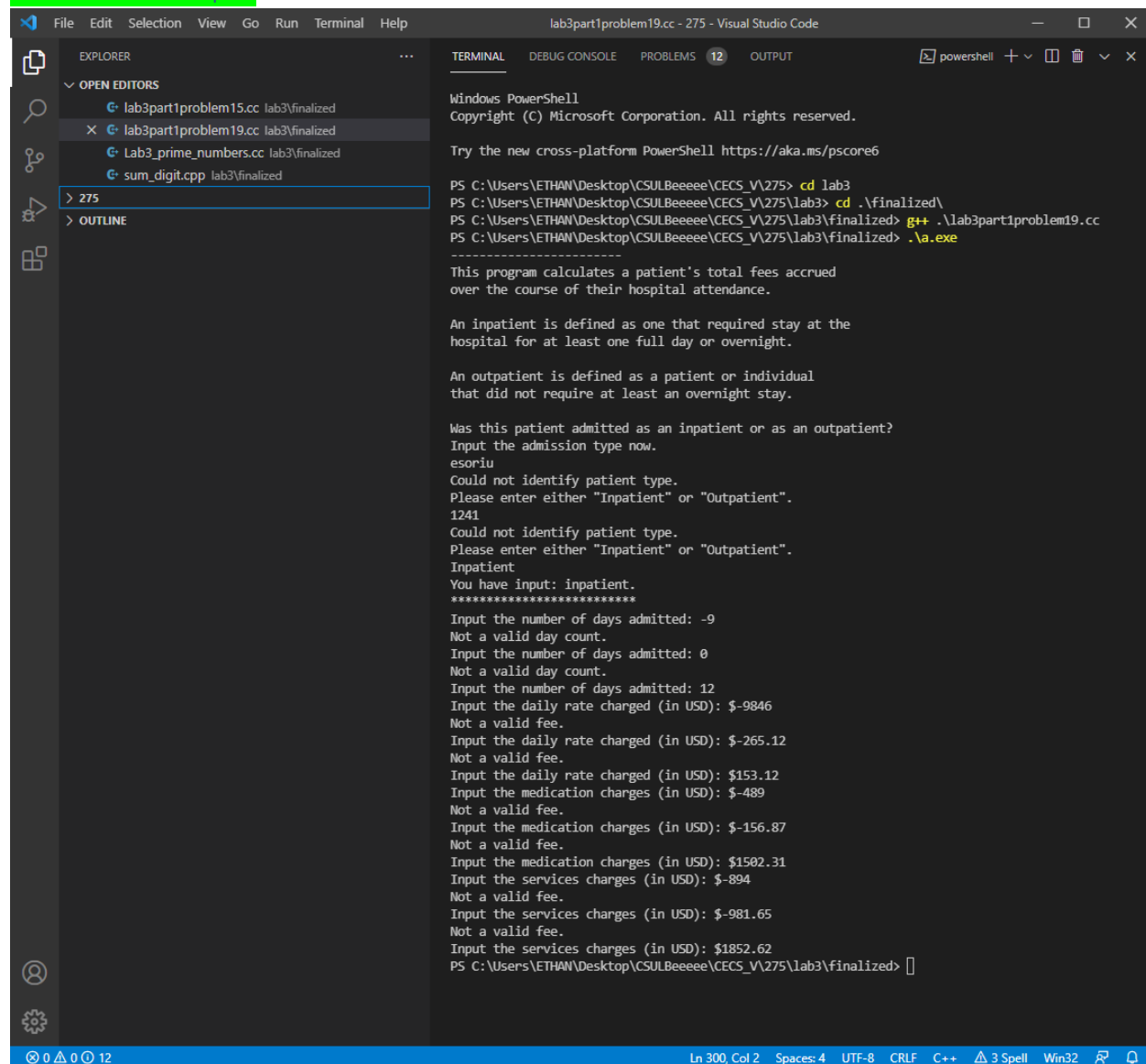
```
289     * Calculates sum of fees input by user, assuming they are an outpatient.
290     * Overloaded function case 2 defined by problem 15.
291     * Differentiated by use of only 2 parameters.
292     *
293     * @param medication aggregate cost of medications used on patient
294     * @param services fees for hospital equipment upkeep/use
295     */
296    double billGenerator(double medication, double services){
297        // simple sum of two parameters
298        double totalCharges = medication + services;
299        return totalCharges;
300    }
```

File   Edit   Selection   View   Go   Run   Terminal   Help   lab3part1problem19.cc - 275 - Visual Studio Code

EXPLORER

∨ OPEN EDITORS
    lab3part1problem15.cc lab3\finalized
    × lab3part1problem19.cc lab3\finalized
    Lab3_prime_numbers.cc lab3\finalized
    sum_digit.cpp lab3\finalized
> 275
> OUTLINE

TERMINAL   DEBUG CONSOLE   PROBLEMS 12   OUTPUT

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275> cd lab3
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3> cd .\finalized\
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3\finalized> g++ .\lab3part1problem19.cc
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3\finalized> .\a.exe
------------------------
This program calculates a patient's total fees accrued
over the course of their hospital attendance.

An inpatient is defined as one that required stay at the
hospital for at least one full day or overnight.

An outpatient is defined as a patient or individual
that did not require at least an overnight stay.

Was this patient admitted as an inpatient or as an outpatient?
Input the admission type now.
esoriu
Could not identify patient type.
Please enter either "Inpatient" or "Outpatient".
1241
Could not identify patient type.
Please enter either "Inpatient" or "Outpatient".
Inpatient
You have input: inpatient.
***************************
Input the number of days admitted: -9
Not a valid day count.
Input the number of days admitted: 0
Not a valid day count.
Input the number of days admitted: 12
Input the daily rate charged (in USD): $-9846
Not a valid fee.
Input the daily rate charged (in USD): $-265.12
Not a valid fee.
Input the daily rate charged (in USD): $153.12
Input the medication charges (in USD): $-489
Not a valid fee.
Input the medication charges (in USD): $-156.87
Not a valid fee.
Input the medication charges (in USD): $1502.31
Input the services charges (in USD): $-894
Not a valid fee.
Input the services charges (in USD): $-981.65
Not a valid fee.
Input the services charges (in USD): $1852.62
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3\finalized>
```

Ln 300, Col 2   Spaces: 4   UTF-8   CRLF   C++   3 Spell   Win32

```
Services Rendered
******************
Patient was admitted for 12 days.
Daily Rate...............: $153.12
     Daily Total..........: $1837.44
Medication Fees..........: $1502.31
Services Fees............: $1852.62

Final charges incurred...: $5192.37
```

File   Edit   Selection   View   Go   Run   Terminal   Help

EXPLORER

∨ OPEN EDITORS
    lab3part1problem15.cc lab3\finalized
  ✕ lab3part1problem19.cc lab3\finalized
    Lab3_prime_numbers.cc lab3\finalized
    sum_digit.cpp lab3\finalized
> 275
> OUTLINE

TERMINAL   DEBUG CONSOLE   PROBLEMS 12   OUTPUT

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275> cd lab3
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3> cd .\finalized\
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3\finalized> g++ .\lab3part1problem19.cc
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3\finalized> .\a.exe
----
This program calculates a patient's total fees accrued
over the course of their hospital attendance.

An inpatient is defined as one that required stay at the
hospital for at least one full day or overnight.

An outpatient is defined as a patient or individual
that did not require at least an overnight stay.

Was this patient admitted as an inpatient or as an outpatient?
Input the admission type now.
iowetu
Could not identify patient type.
Please enter either "Inpatient" or "Outpatient".
-13523
Could not identify patient type.
Please enter either "Inpatient" or "Outpatient".
Outpatient
You have input: outpatient.
****
Input the medication charges (in USD): $-94156
Not a valid fee.
Input the medication charges (in USD): $-8416.16
Not a valid fee.
Input the medication charges (in USD): $4512.16
Input the services charges (in USD): $-984
Not a valid fee.
Input the services charges (in USD): $-654135.13
Not a valid fee.
Input the services charges (in USD): $1568.49
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3\finalized>

Ln 300, Col 2   Spaces: 4   UTF-8   CRLF   C++   ⚠ 3 Spell   Win32

```
Services Rendered
******************
Medication Fees..........: $4512.16
Services Fees...........: $1568.49

Final charges incurred...: $6080.65
```

Part 2

```cpp
/*
 * Prime number finder
 * CECS 275 - Fall 2021
 * @author Rodrigo Becerril Ferreyra
 * @author Ethan Hua
 * @version 2
 */
#include <fstream> // for file output

bool isPrime(int);

int main()
{
    std::ofstream outFile("prime_numbers.txt");
    // Problem 23 asks for the prime numbers from 1 to 100
    for(int i = 1; i <= 100; ++i)
    {
        if(isPrime(i))
        {
            outFile << i << " ";
        }
    }
    outFile.close();

    return 0;
}

/**
 * Determines whether or not the input is prime
 * using the brute-force trial division algorithm.
 * @param numToTest number that is either prime or composite
 * @return true if numToTest is prime, false if composite
 */
bool isPrime(int numToTest)
{
    /*
    Explanation of operation: the program starts by dividing numTo
Test
    by every whole number starting from 2. If numToTest is divisib
le by any
    number, then it is composite. If there is no number which doe
s not give
    a remainder, then numToTest is prime.
    */

    // 1 is neither prime nor composite.
    // Anything less is undefined.
    if(numToTest < 2) return false;

    for(int divisor = 2; divisor < numToTest; ++divisor)
    {
        if(numToTest % divisor == 0)
            return false;
    }
    return true;
}
```

prime_numbers - Notepad

File   Edit   Format   View   Help

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

Ln 1, Col 1   100%   Windows (CRLF)   UTF-8

Part 3

```cpp
/*
 * Sum of All Digits Calculator
 * CECS 275 - Fall 2021
 * @author Rodrigo Becerril Ferreyra
 * @author Ethan Hua
 * @version 4
 */
#include <iostream>

int sumOfAllDigits(int);

int main()
{
    int user_value, total, old_value;
    // The following while loop runs until the user inputs a positive integer.
    while(true)
    {
        std::cout << "Please enter a positive integer: ";
        std::cin >> user_value;
        if(user_value <= 0)
        {
            std::cout <<
"Positive integers are whole numbers greater than 0. "
                      << "Please try again." << std::endl;
            continue;
        }

        std::cout << "You have entered " << user_value << "." << std::endl;

        int total = sumOfAllDigits(user_value);

        std::cout << "The sum of all the digits in " << user_value << " is "
                  << total << "." << std::endl;
    }
}

/**
* Adds the sum of the digits of the input and returns that sum.
* @param numToSum the number whose digits will be summed
* @return the sum of all digits
*/
int sumOfAllDigits(int numToSum)
{
    int total = 0;

    while(numToSum != 0)
    // numToSum being 0 means that the variable was one digit in this
    //    iteration, and that there are no more digits to total.
    {
        // The expression "% 10" returns the most insignificant digit
        //    (in decimal): for example, 90121 % 10 returns 1.
        total += numToSum % 10;
        // Due to the properties of integer division, dividing by 10
        //    removes the variable's least significant digit.
        numToSum /= 10;
    }

    return total;
}
```
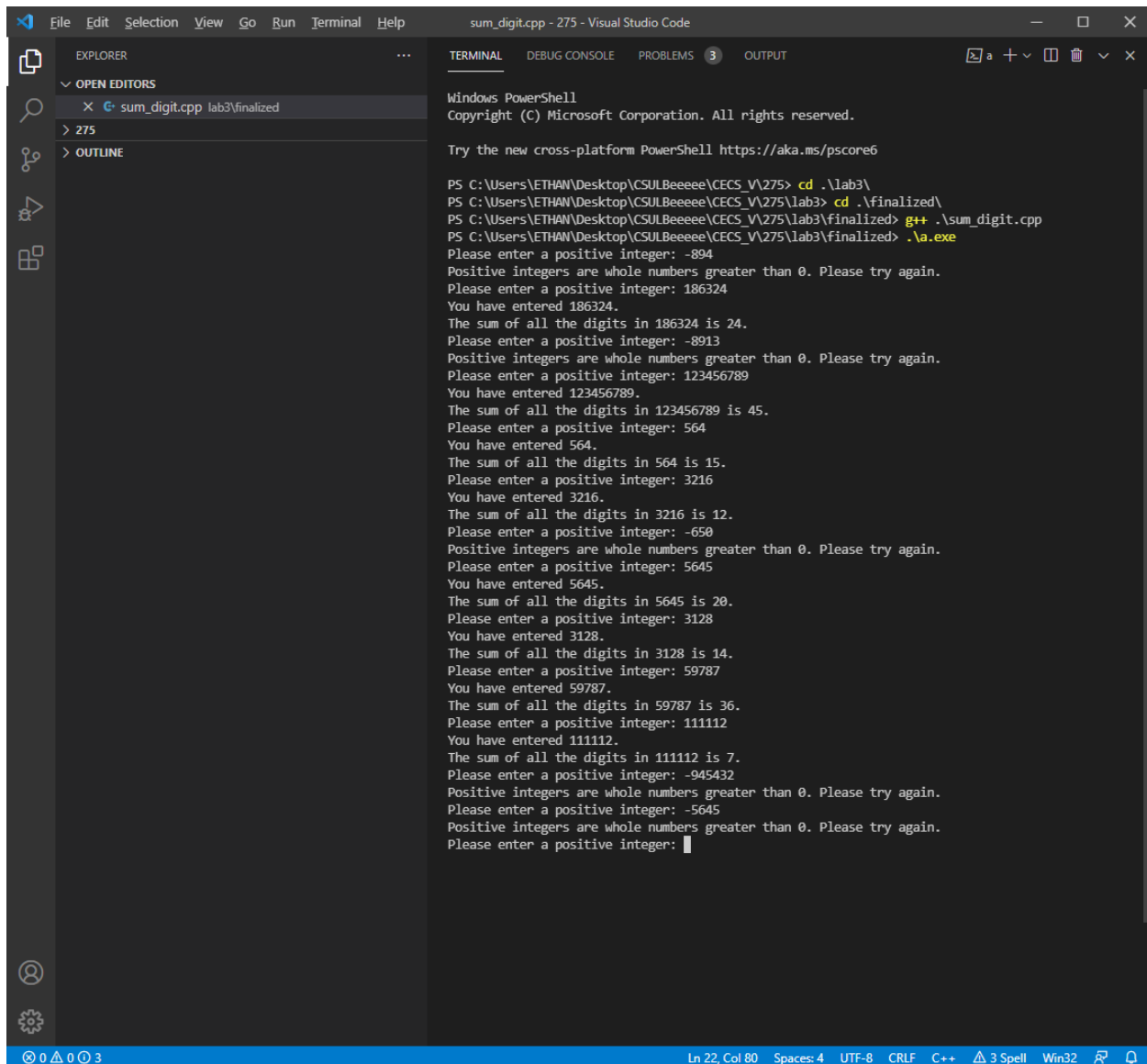
File Edit Selection View Go Run Terminal Help

EXPLORER

∨ OPEN EDITORS
  ✕ C⁺ sum_digit.cpp lab3\finalized
> 275
> OUTLINE

TERMINAL   DEBUG CONSOLE   PROBLEMS 3   OUTPUT

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275> cd .\lab3\
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3> cd .\finalized\
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3\finalized> g++ .\sum_digit.cpp
PS C:\Users\ETHAN\Desktop\CSULBeeeee\CECS_V\275\lab3\finalized> .\a.exe
Please enter a positive integer: -894
Positive integers are whole numbers greater than 0. Please try again.
Please enter a positive integer: 186324
You have entered 186324.
The sum of all the digits in 186324 is 24.
Please enter a positive integer: -8913
Positive integers are whole numbers greater than 0. Please try again.
Please enter a positive integer: 123456789
You have entered 123456789.
The sum of all the digits in 123456789 is 45.
Please enter a positive integer: 564
You have entered 564.
The sum of all the digits in 564 is 15.
Please enter a positive integer: 3216
You have entered 3216.
The sum of all the digits in 3216 is 12.
Please enter a positive integer: -650
Positive integers are whole numbers greater than 0. Please try again.
Please enter a positive integer: 5645
You have entered 5645.
The sum of all the digits in 5645 is 20.
Please enter a positive integer: 3128
You have entered 3128.
The sum of all the digits in 3128 is 14.
Please enter a positive integer: 59787
You have entered 59787.
The sum of all the digits in 59787 is 36.
Please enter a positive integer: 111112
You have entered 111112.
The sum of all the digits in 111112 is 7.
Please enter a positive integer: -945432
Positive integers are whole numbers greater than 0. Please try again.
Please enter a positive integer: -5645
Positive integers are whole numbers greater than 0. Please try again.
Please enter a positive integer:
```

⊗ 0 ⚠ 0 ⓘ 3   Ln 22, Col 80   Spaces: 4   UTF-8   CRLF   C++   ⚠ 3 Spell   Win32