CS 132

Project 2: "Phonepad Puzzle"

Name:

Each week on ***National Public Radio's Weekend Edition-Sunday*** there is a puzzle challenge, presented by Will Shortz, puzzle editor for the *New York Times*. The challenge for January 11, 2004 was:

This numerical puzzle originates from the Turkish Intelligence Foundation sent to Will Shortz by Turkish puzzler Metin Orsel. Start with a certain number on a telephone keypad. Move to another number adjacent to it horizontally, vertically or diagonally. Then move to another number adjacent to that one horizontally, vertically or diagonally. And keep doing this until you have 9 digits. All the digits must be different. That is, you'll use 9 of the 10 digits on the phone including zero (0) and the path must not cross itself. When you're done, the number formed by the first 3 digits plus the number formed by the next 3 digits will equal the/ number formed by the last 3 digits. There's only one path that reaches this solution. What is it?

1	**2**	**3**
4	**5**	**6**
7	**8**	**9**
	0	

It turns out that there are actually two correct solutions. If you did not have the restriction of no crossed paths, there would be three solutions.

Your task is to write a C++ program which uses **recursion** to find the three possible solutions. You can then decide on your own which solution should be discarded for having crossing paths. You can accomplish this task in the following manner:

- create a const *10X9* array that represents all of the adjacencies for each digit on the phonepad. That is, each row will start with a digit(0-9), the first row will have in it:
  {0,7,8,9,-1,-1,-1,-1,-1}
  where the 0 represents the 0 digit on the phone, the 7, 8 and 9 represent the digits adjacent to the 0, and the -1's are used as fill.
  In a like manner, the row representing the 5 will contain:
  {5,1,2,3,4,6,7,8,9}

  Alternately, you can have a *10X10* array of booleans, where the row number represents a digit, and the column number represents possible adjacent digits. Any single element is set to true ***iff*** the digit represented by the column is adjacent to the digit represented by the row.
- in main, create the array one_answer[9], which holds the 9 digits of the answer.
- for each of the digits (0-9) set one_answer[*0*] to the digit and call the recursive function nextdigit.

It would probably be a good idea to define the following four functions:

- void nextDigit(int oneAns[], int place)
  where oneAns is the number array and place is a number representing the index of which digit is the last to have been added to the array.

  this function calls repeat and, if any digits are repeats of previous digits, exits the function.
  If all of the digits have been calculated, calls isSolution to see if the number works as a solution.
  Otherwise, it goes through each number in the adjacency array for the current digit, adds it in the next place in the array, and makes a **recursive** call.

- bool repeat(int oneAns[], int place)
  where oneAns is the number array and place is a number representing the index of which digit is being added to the array.

  this function goes through the digits up to *place* and checks to see if any is the same as the digit **at** *place*. It immediately returns **true** if there is a match. Return a **false** if no match is found.

- bool isSolution(int oneAns[])
  where oneAns is the number array .

  this function takes the first three digits as an **int**, adds that to the second three digits as an **int** and compares the sum to the last three digits as an **int**. It returns **true** ***iff*** they match.

- void print(int oneAns[])
  where oneAns is the number array. this method prints out the numbers in the form ***nnn* + *nnn* = *nnn***.

Points Possible: 100

Extra Credit: 10 points

Add in and utilize a function to check if the solution has crossed any paths.
Hint: Any two numbers that are adjacent diagonally add up to an even number;
Any two numbers that are adjacent vertically or horizontally add up to an odd number.
Paths can only cross diagonally, and any two that **do** cross have the same sum. *e.g.,* 1 + 5 = 2 + 4

Deliverables:

Physical:

- *The project should be turned in inside a clear plastic Deluxe Locking Project File Folder DOCU Manager or equivalent. This folder should have a simple flap to hold paper in place--NO buttons, strings, velcro, etc.* Pages should be in order, **not** stapled.
- Assignment Sheet *(this page),* with your **name** written on it, as a cover sheet.
- Printed Source Code with Comments *(including heading blocks. Describe parameters, no line wrapping)*
- Sample Output *(printed)*
- a simple test plan including explanations of any discrepancies and reasons for each test. Show **ALL** values output as well as **ALL** expected output. Make sure it is clear which of the 3 solutions is invalid.

Electronic:

- Allfiles ***.h***, ***.cpp, and .exe****(Release Version)files,* ***zipped*** together. **Do not** use ***rar*** or any archive format other than ***zip***. Rename the file: "<*YourNames*>_p2.zip".
- Submit this single ***zip*** file by going to **Canvas**, select this class, select the **Assignment** tab on the left, select the ***Assignment 1***, select the **submit Assignment** tab at the top right, find the file, and **Submit**.

Due: Tuesday, February 18, 2014, 9:30 am *(beginning of class)*
Resubmittal due: Two weeks from the day the project is returned in class (*beginning of class*).