

Project 1 Algorithm:

Step 1: Start: Create a cpp file. Use appropriate libraries `<iostream>` `<string>` `<vector>` `<string>`.

Step 2: In Main function declare the int vectors for N1, N2, N3, N4, N5, N6, N7, N8, N9, N10

Step 3: Declare unsigned long long integer variable 't' and set it equal to 0

Step 4: Use a For loop using an unsigned integer variable 'n' initialized at 1 increment till the n is less than 10 ($n < 10$); n increments to 9. Within the for loop write an if statement where if n is evenly divisible by 1 pushback the value for n into the vector 'N1'; this process is redundant and thus optional for 'N1'.

Step 5: Use 'N1' to generate the next set of 2 digit permutations. Use a for loop using an integer variable 'n' initialized at 1 with 'n' being less than 10 and incremented to 9. Number n is used in generating new elements for the next variable. Within the loop declare a char variable 'cn' and set it equal to the variable 'n' converted to a char. Then declare a nested for loop using an unsigned integer (necessary for a vectors) variable 'i' and set it equal to 0 with 'i' and 'i' being incremented to less than the size of the vector, 'N1'. Variable 'i' keeps track of position in vector. Convert the vector value at the position 'i' for 'N1' to a string using the 'to_string' function. Create another nested for loop to iterate through the chars of the converted string value for N1 at position 'i'. Declare a conditional statement that compare the the vector element to char 'cn' to avoid any repeating numbers in a permutation solution. Declare a new integer variable "num" and set it equal to the integer value of 'N1' at position 'i'. To generate a new number multiplied 'num' by 10 and add the variable 'n'. Write a conditional if statement that checks to see if the value of "num" is evenly divisible by 2. If so pushback the value to the vector 'N2'.

Step 6: Use a for loop with an unsigned integer variable 'i' set equal to 0 with 'i' being less than the size of the vector N2 and incremented. Within the loop declare an integer variable "num" and set it equal to . Declare another for loop that uses an unsigned integer variable 'n' set equal to 1 where 'n' is less than 10 and incremented 9 times. Declare a char variable 'cn' and set it equal to the 'n' variable converted to a char. Declare a string "str" and assign the value of N2 at position 'i' converted to a string. Declare a conditional statement that compare the the vector elements to char in both positions to avoid any repeating numbers in a permutation solution as well as checks to see if the value of "num" is evenly divisible by 3. Declare a new integer variable "num" and set it equal to the result of the value of N2 at position 'i' multiplied by 10 and added with the value of 'n'. If so pushback the value to the vector N3.

Step 7: Use a for loop with an unsigned integer variable 'i' set equal to 0 with 'i' being less than the size of the vector N3 and incremented. Within the loop declare an integer variable "num" and set it equal to . Declare another for loop that uses an unsigned integer variable 'n' set equal to 1 where 'n' is less than 10 and incremented 9 times. Declare a char variable 'cn' and set it equal to the 'n' variable converted to a char. Declare a string "str" and assign the value of N3 at position 'i' converted to a string. Declare a conditional statement that compare the the vector elements to char in the three positions to avoid any repeating numbers in a permutation solution as well as checks to see if the value of "num" is evenly divisible by 4. Declare a new integer variable "num" and set it equal to the result of the value of N3 at position 'i' multiplied by 10 and added with the value of 'n'. If so pushback the value to the vector N4.

Step 8: Use a for loop with an unsigned integer variable 'i' set equal to 0 with 'i' being less than the size of the vector N4 and incremented. Within the loop declare an integer variable "num" and set it equal to . Declare another for loop that uses an unsigned integer variable 'n' set equal to 1 where 'n' is less than 10 and incremented 9 times. Declare a char variable 'cn' and set it equal to the 'n' variable converted to a char. Declare a string "str" and assign the value of N4 at position 'i' converted to a string. Declare a conditional statement that compare the the vector elements to char in all four positions to avoid any repeating numbers in a permutation solution as well as checks to see if the value of "num" is evenly divisible by 5. Declare a new integer variable "num" and set it equal to the result of the value of N4 at position 'i' multiplied by 10 and added with the value of 'n'. If so pushback the value to the vector N5.

Step 9: Use a for loop with an unsigned integer variable 'i' set equal to 0 with 'i' being less than the size of the vector N5 and incremented. Within the loop declare an integer variable "num" and set it equal to . Declare another for loop that uses an unsigned integer variable 'n' set equal to 1 where 'n' is less than 10 and incremented 9 times. Declare a char variable 'cn' and set it equal to the 'n' variable converted to a char. Declare a string "str" and assign the value of N5 at position 'i' converted to a string. Declare a conditional statement that compare the the vector elements to char in all 5 positions to avoid any repeating numbers in a permutation solution as well as checks to see if the value of "num" is evenly divisible by 6. Declare a new integer variable "num" and set it equal to the result of the value of N5 at position 'i' multiplied by 10 and added with the value of 'n'. If so pushback the value to the vector N6.

Step 10: Use a for loop with an unsigned integer variable 'i' set equal to 0 with 'i' being less than the size of the vector N6 and incremented. Within the loop declare an integer variable "num" and set it equal to . Declare another for loop that uses an unsigned integer variable 'n' set equal to 1 where 'n' is less than 10 and incremented 9 times. Declare a char variable 'cn' and set it equal to the 'n' variable converted to a char. Declare a string "str" and assign the value of N6 at position 'i' converted to a string. Declare a conditional

statement that compare the the vector elements to char in all 6 positions to avoid any repeating numbers in a permutation solution as well as checks to see if the value of "num" is evenly divisible by 7. Declare a new integer variable "num" and set it equal to the result of the value of N6 at position 'i' multiplied by 10 and added with the value of 'n'. If so pushback the value to the vector N7.

Step 11: Use a for loop with an unsigned integer variable 'i' set equal to 0 with 'i' being less than the size of the vector N7 and incremented. Within the loop declare an integer variable "num" and set it equal to . Declare another for loop that uses an unsigned integer variable 'n' set equal to 1 where 'n' is less than 10 and incremented 9 times. Declare a char variable 'cn' and set it equal to the 'n' variable converted to a char. Declare a string "str" and assign the value of N7 at position 'i' converted to a string. Declare a conditional statement that compare the the vector elements to char in all 7 positions to avoid any repeating numbers in a permutation solution as well as checks to see if the value of "num" is evenly divisible by 8. Declare a new integer variable "num" and set it equal to the result of the value of N7 at position 'i' multiplied by 10 and added with the value of 'n'. If so pushback the value to the vector N8.

Step 12: Use a for loop with an unsigned integer variable 'i' set equal to 0 with 'i' being less than the size of the vector N8 and incremented. Within the loop declare an integer variable "num" and set it equal to . Declare another for loop that uses an unsigned integer variable 'n' set equal to 1 where 'n' is less than 10 and incremented 9 times. Declare a char variable 'cn' and set it equal to the 'n' variable converted to a char. Declare a string "str" and assign the value of N8 at position 'i' converted to a string. Declare a conditional statement that compare the the vector elements to char in all 8 positions to avoid any repeating numbers in a permutation solution as well as checks to see if the value of "num" is evenly divisible by 9. Declare a new integer variable "num" and set it equal to the result of the value of N8 at position 'i' multiplied by 10 and added with the value of 'n'. If so pushback the value to the vector N9.

Step 13: Declare a new integer variable 't' and set it equal to the result of the value of N9 at position 0 multiplied by 10 and added with the value of 0. Then use a conditional if statement to check to see if 't' is every divisible by 10. If so pushback the value to the vector N10.

Step 14: Write an output statement for each count ranging 1-7. For example "Count for 1 digit numbers" and output the size of vector N1....N7 for their corresponding Count statements.

Step 15: Create a for loop that uses an integer 'i' set equal to 0 where 'i' is less than the size of the vector N8 and 'i' is incremented. Within the loop output all values stored in N8 with a new line appearing after each iteration of the loop.

Step 16: Declare the output statement "All possible solutions for 10 digit numbers: ".
Next declare an output statement to output the values stored in the vector N10 at position 0.

Step 17: To generate hexadecimal permutations create a new void function with. Inside the function declare an unsigned long long int vector N1 and 15 string vectors: N2, N3, N4, N5, N6, N7, N8, N9, N10, NA, NB, NC, ND, NE, NF. These vectors store permutations of 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 digits respectively.

Hex:

Step 1: Initialize a variable 'n' at 1. Increment the variable till 'n' is less than 10; 'n' increments to 9. Insert the variable into the vector 'N1' using the "push_back" method.

Step 2: Use 'N1' to generate 'N2'. Loop through variable 'N1'. Declare one integer 'num' variable to perform mathematical operations on elements of 'N1' to generate 'N2'. Declare two string 'str' and 'snum'; one for storing elements of the 'N1' using 'to_string' method from string class, and the other for storing elements of 'N2'. Create a nested for loop, declare a variable 'n' initialized at 1 and incremented to 9. Declare a char 'cn' to store int 'n' in a char form. Use a conditional for all indices of the strings of 'N1' compare to char 'cn'. If the char 'cn' is not already in the previous number execute the code block under the conditional. Inside the code block we generate a number store it to 'num' by multiplying the current element of 'N1' in the loop by 10 and summing int 'n' into the result, this generate a permutation of 2 digit numbers. Use 'to_string' method on the permutation and store it in the string 'snum' Convert the numbers to an int of base 16 using method 'stoi' and store it back to num. Use a conditional to check for even visibility by 2, if it passes the condition append it to the vector using 'push_back' method.

Step 3: Use 'N2' to generate 'N3'. Loop through variable 'N2'. Declare one integer 'num1' variable to perform mathematical operations on elements of 'N2' to generate 'N3' Use 'stoi' method to convert string to a decimal int and store it in num 1. Declare two string 'str' and 'snum'; one for storing elements of the 'N2', and the other for storing elements of 'N3'. Create a nested for loop, declare a variable 'n' initialized at 1 and incremented to 9. Declare a char 'cn' to store int 'n' in a char form. Use a conditional for all indices of the current strings from 'N2' compare to char 'cn'. Use a conditional to compare chars of 'str' to char 'cn'. If the char 'cn' is not already in the string 'str' execute the code block under the conditional. Inside the code block we generate the a new number 'num' by multiplying 'num1' by 10 and summing int 'n' into the result, this generate a permutation of 3 digit numbers. Use 'to_string' method on the permutation and store it in the string for 'snum'. Convert the numbers to an int of base 16 using method 'stoi' and store it back to num. Use a conditional to check for even visibility by 3,

if it passes the condition append it to the vector using 'push_back' method. Repeat step 3 till N10.

Step 4: Use 'N10' to generate 'NA'. Use a for loop to iterate through the vector 'N10'. Store elements of 'N10' in a string 'str'. Create a nested for loop to loop through the string. Use insert method to insert 'A' at the iterator position in the string. Convert the string to a base 16 in using 'stoi' method. Use a conditional to check for even visibility by 11; if it passes true append the string to the vector 'NA' using 'push_back' method. Repeat process till 'NF'.

Step5: display counts of permutations of digits 1-7 and display the outputs of the permutations digits 8-16. If the permutation doesn't exist display message: "There are no solutions for [n] digit number".