You work for the Central Intelligence Service, in Department 4050 (AKA CIS4050). Your job is to write software and you have been selected for your first assignment.

Some of the safe boxes the department has around the world to keep its secrets use a simple keypad just like the one in Figure 1

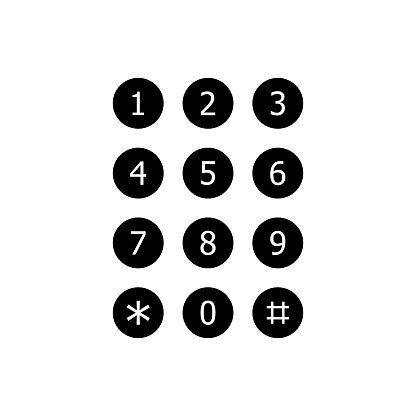


Figure 1: CIS safe box keypad

Each agent has a 900 number (like the one you got from MSU Denver). To get the combination for the box, each agent must enter the 6 digits after the 900 of their ID using a specific set of instructions.

To enter each code the agent must look for her digit in a sheet of instructions that are unique for each agent. The instructions help the agent to navigate the keypad to enter a digit. The open arrows in the instructions indicate directions. For example, **^** indicates “up”. Each set of instruction starts from the key corresponding to the set of instructions.

Assume that agent Smart’s sheet of instructions is the following:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Code 0 | Code 1 | Code 2 | Code 3 | Code 4 | Code 5 | Code 6 | Code 7 | Code 8 | Code 9 |
| <^v> | >>^^ | <vv> | vv^< | <v^< | <^>v | vv>^ | ^>^^ | <><< | ^<>> |

Table 1: Example of set of instructions

Now, agent Smart ID is 900123456. He will use Code 1, Code 2, Code 3, Code 4, Code 5, and Code 6.

Here is how agent Smart will proceed with the Code 1:

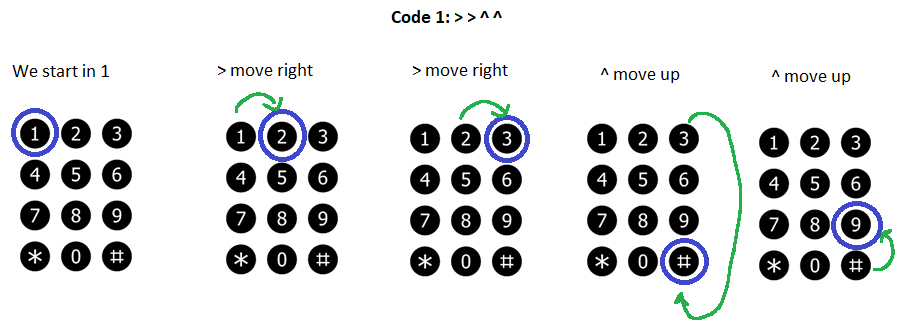


Figure 2: Representation of Agent Smart first digit solution

Therefore, the agent must press “9”.

Here is how agent Smart will proceed with the Code 2:

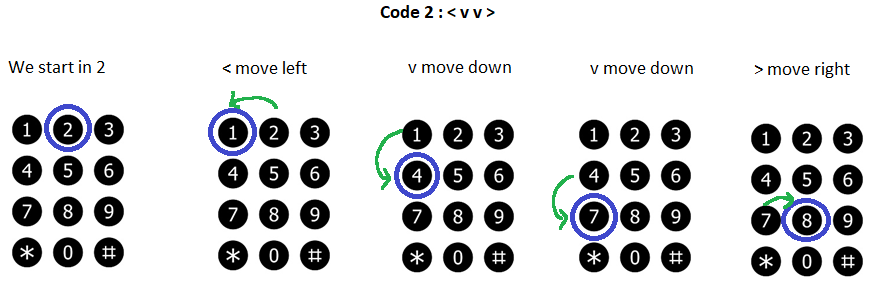


Figure 3: Representation of agent Smart's second digit solution

Therefore, the agent must press “8”.

After following all the procedures, agent Smart will press:

Code 1: Press 9

Code 2: Press 8

Code 3: Press 5

Code 4: Press 5

Code 5: Press 5

Code 6: Press 7

Be aware that an agent’s final password may include “\*” or “#”.

**First Part of the Decoding Process (70%)**

The file “codes.txt” has all the codes for all the agents in CIS4050. Your job is to create a program that decode the password to enter given an agent name and 900 number. You must look first your list of codes in the document (First name, initial Last Name) and code the solution.

You must deliver for this section a python program that solves the problem and the final answer (6 digits/symbols) for any agent’s set of instructions.

Rules:

* I must be able to run your code. No exceptions.
* Your solution must be universal (your program must be able to get the codes of any agent).
* Your solution MUST read the codes from the file codes.txt.
* The system ONLY will request agent’s credentials.

Once the first door is opened, the agent finds a second door with a set of switches (see Figure 4). To open the second door, the agent needs to set the switches on the right position.

To go over this part, an agent needs the 6 symbols (they may be digits and/or symbols) obtained from the first part of the decoding, as well as her #900 digits. With these two sets, the agent will proceed with the following steps:

1. Express each symbol from the first decoding section as the binary equivalent in [ASCII](https://en.wikipedia.org/wiki/ASCII). Use 8 bits.
2. Express each digit from the agent’s las 6 digits of her 900# as the binary equivalent in ASCII. Use 8 bits.
3. The binary expressions will be used in a “[bracket-like](https://en.wikipedia.org/wiki/Bracket_(tournament))” set of [logic gate operations](https://en.wikipedia.org/wiki/Logic_gate). The operations are [bitwise](https://en.wikipedia.org/wiki/Bitwise_operation).
4. The result will be used to set an 8-bit switch similar to the one in Figure .

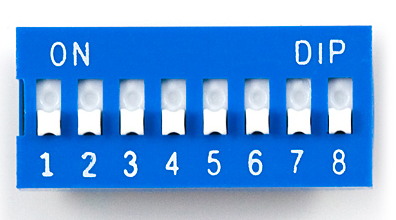


Figure 4: 8-bit switch used to enter the second code

The bracket has the following form:

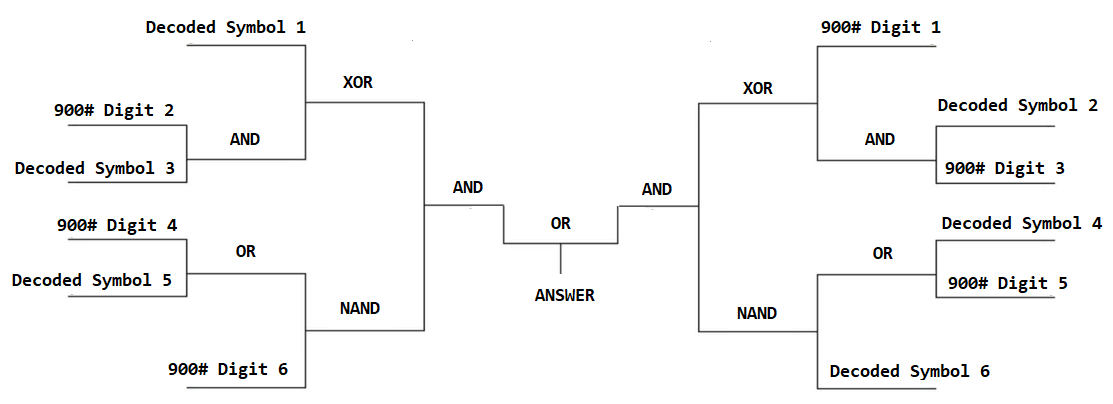


Figure 5: Bracket-like logical operation to get the second code

For example, after agent Smart solved the first section, his decoded number was 985557. As it was indicated before his 900# is 900123456. The equivalent of the password in ASCII for his decoded number and 900# digits are shown in Table .

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Decoded Symbol 1 | Decoded Symbol 2 | Decoded Symbol 3 | Decoded Symbol 4 | Decoded Symbol 5 | Decoded Symbol 6 |
|  | 9 | 8 | 5 | 5 | 5 | 7 |
| ASCII | 00111001 | 00111000 | 00110101 | 00110101 | 00110101 | 00110111 |

Table 2: Ascii equivalent of agent Smart #900 numbers

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 900# Digit 1 | 900# Digit 2 | 900# Digit 3 | 900# Digit 4 | 900# Digit 5 | 900# Digit 6 |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| ASCII | 00110001 | 00110010 | 00110011 | 00110100 | 00110101 | 00110110 |

Table 3: ASCII equivalent of agent Smart Decoded Symbols and #900 numbers

After using all the binary codes in the bracket, the final code is **00001001.** See Figure 6.

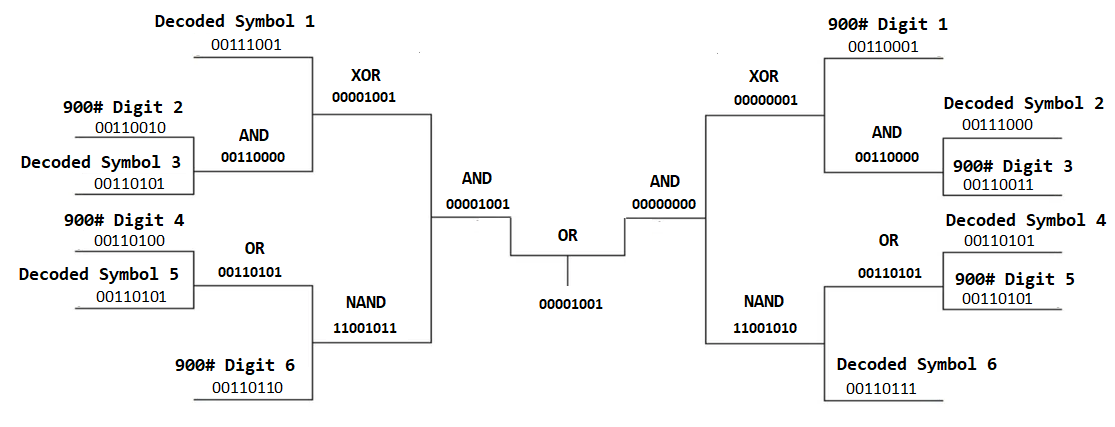


Figure 6: Solution for agent Smart second code

**Second part of the Decoding Process (30%)**

Your job is to create a program that gets the second code given the first code and an agent’s #900.

You must deliver for this section a python program that solves the problem and the final answer (8 bits) for any first code and #900 number.

Rules:

* I must be able to run your code. No exceptions.
* Your solution must be universal (your program must be able to get the code of any agent)
* The system ONLY will request agent’s credentials and the code from the first part.

For testing purposes, the following are the results for agent EdgarM using a #900 = **012345**

