

```

{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "markdown",
      "source": [
        "First we need to import the necessary libraries for the problem.\n",
        "\n",
        "Z3 is what we will be using to solve this. If Z3 is not already installed then use: pip install z3-solver\n",
        "\n",
        "I imported the product function from the itertools library just to make the nested loops easier to read (in my opinion). It is
not necessary to solve this problem."
      ],
      "metadata": {
        "id": "p0Das2ovgj_b"
      }
    },
    {
      "cell_type": "code",
      "execution_count": 2,
      "metadata": {
        "id": "hRn27GcYv8zx"
      },
      "outputs": [],
      "source": [
        "from z3 import *\n",
        "from itertools import product"
      ]
    },
    {
      "cell_type": "markdown",
      "source": [
        "Next we input the combinations given."
      ],
      "metadata": {
        "id": "HfIDPaOThDoc"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "A = [2,6,5,7]\n",
        "B = [0,4,1,5]\n",
        "C = [4,2,6,8]\n",
        "D = [1,7,4,9]"
      ],
      "metadata": {

```

```

    "id": "UdfRvxxzPkmz"
  },
  "execution_count": 3,
  "outputs": []
},
{
  "cell_type": "markdown",
  "source": [
    "Lastly, we set up our model.\n",
    "\n",
    "We add the 4 constraints to the model and if the model is satisfiable, we print out the solution. "
  ],
  "metadata": {
    "id": "lEK9oy41hIXh"
  }
},
{
  "cell_type": "code",
  "source": [
    "s = Solver()\n",
    "X = IntVector('x', 4)\n",
    "s += [And(x>=0, x<=9) for x in X]\n",
    "\n",
    "s += sum([If(And(X[i]==A[j], i!=j), 1, 0) for i,j in product(range(len(X)), range(len(A)))] == 2\n",
    "s += sum(If(x==a, 1, 0) for x in X for a in A) == 2\n",
    "\n",
    "s += sum([If(And(X[i]==B[j], i!=j), 1, 0) for i,j in product(range(len(X)), range(len(B)))] == 1\n",
    "s += sum(If(x==b, 1, 0) for x in X for b in B) == 1\n",
    "\n",
    "s += [And(x != c) for x in X for c in C]\n",
    "\n",
    "s += sum([If(And(X[i]==D[j], i==j), 1, 0) for i,j in product(range(len(X)), range(len(D)))] == 2\n",
    "s += sum(If(x==d, 1, 0) for x in X for d in D) == 2\n",
    "\n",
    "if s.check() == sat:\n",
    "    m = s.model()\n",
    "    print([m.evaluate(x) for x in X])"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "9WDIZdbgRbnL",
    "outputId": "3bf35bde-d3e8-41fc-b185-eb20b566d488"
  }
},
  "execution_count": 4,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "[5, 7, 3, 9]\n"
      ]
    }
  ]
}
]
}

```