|  |
| --- |
| **Video** Submit one video in .mp4, .wmv, .avi, or .mov format that demonstrates the running of at least one significant feature of your program. Your video must not exceed 1 minute in length and must not exceed 30MB in size  **Prompt 2a.** Provide a written response or audio narration in your video that:  ●identifies the programming language;  ●identifies the purpose of your program; and  ●explains what the video illustrates.  *(Must not exceed 150 words)* |
| My program, mastermind.py, used Python 3.6.5. The purpose of my program is to simulate the game “mastermind”. The video shows a quick overview of my code, and a demonstration of the how the game works. |

|  |
| --- |
| **2b.** Describe the incremental and iterative development process of your program, focusing on two distinct points in that process. Describe the difficulties and / or opportunities you encountered and how they were resolved or incorporated. In your description clearly indicate whether the development described was collaborative or independent. At least one of these points must refer to independent program development. *(Must not exceed 200 words)* |
| I used an incremental process while calculating the number of attempts/guesses the user made before they correctly got the sequence. I used an iterative process while checking the position of the guess to check each position of the list. I ran into an issue while having my friends test my program because they entered in letters/more than 4 characters to break it. To solve this, I added a try/catch statement as well as an if statement to check if the list had 4 characters. My program was developed 100% independent. |

|  |
| --- |
| **2c.** Capture and paste a program code segment that implements an algorithm (marked with an **oval** in **section 3**) and that is fundamental for your program to achieve its intended purpose. This code segment must be an algorithm you developed individually on your own, must include two or more algorithms, and must integrate mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program. *(Must not exceed 200 words)* |
| Code Segment |
| def game():  time = 1  lst = list(map(str, [random.randint(0, 3) for \_ in range(4)]))  # print(lst)  guess = [""] \* 4  while guess != lst:  guess = get\_guess()  if guess == lst:  print("Congrats! You got it correct after {} attempt(s)!".format(time))  exit()  correct = 0  for pos in range(4):  if guess[pos] == lst[pos]:  correct += 1  print("Incorrect, you got {} correct".format(correct))  if time % 4 == 0:  resp = input("Would you like a hint? (Y/N)\n")  if str(resp[0]).upper() == "Y":  loc = random.randint(0, 3)  print("Position {} is {}".format(loc, lst[loc]))  time += 1 |
| Written Response |
| One algorithm I have in my program checks if the user’s guess is the correct answer. If it isn’t, it will continue to my other code/algorithms. If it is, it will end the program after congradulating them and telling them how many attempts they took.  Later in the program, I have another algoritms interates through each position of the user’s guess. This is to tell them how many positions they got correct. |

|  |
| --- |
| **2d.** Capture and paste a program code segment that contains an abstraction you developed individually on your own (marked with a **rectangle** in **section 3**). This abstraction must integrate mathematical and logical concepts. Explain how your abstraction helped manage the complexity of your program. *(Must not exceed 200 words)*  *(Must not exceed 250 words)* |
| Code Segment |
| def get\_guess():  try:  guess = list(input("Enter your guesses (4 numbers, no spaces, 0 through 3)\n"))  if len(guess) != 4:  print("You didn't enter 4 numbers")  return get\_guess()  else:  return guess  except (KeyboardInterrupt, IndexError, ValueError):  print("Invalid input!")  return get\_guess() |
| Written Response |
| This abstraction helped my code by making it easier to understand, as well as making it more organized. It’s more intuitive to have this piece of code by itself for readability. Additionally, it allows me to have it call upon itself so that it can form a recursion. If this was in the `game` funcion, it would be much more difficult to implement. |