A1 Q4_Main Program A2 edited train.py A1 edited q4.py get_game_menu() train_or_not(player_name) A2 Q2 game.py Terminator start Start intro_travel_to_camp() from q4 import all functions except main user_input: Sound_hron() Start start from train.py import main Start travel status = Set <u>e flag</u> to False. return a string contain the game menu intro_travel_to_camp() which includes change cheese Print introductions and instructions from Larry Instruct the user to sound Prompt the user to start / the horn by typing "yes" Call intro_travel_to_camp() and Call the input() function with the training or skip store the result in *travel_status* provided prompt and store the Call user_input() function with a prompt to return 'Cardboard result in the user_input variable. travel_statustravel to the Meadow and store the return and Hook Trap' Terminator value to be *travel_status* Call user_input() function with a prompt that ask if user want to arm or not and Call main() from the return 'Cardboard if user store the return value into *horn_input* train module and store chooses to (travel_status)cheddar = Check if the return value in *trap* check if user_input is equal <u>e flag</u> menu(player_name, trap) setup_trap() **(travel_status** is) to the ASCII character with True or not code 27 (Escape check if **(horn_input** is Print the Meadow start description Call menu(player_name, if **cheddar** is _True_<mark>→ return False</mark> while True Call menu(player_name, trap[0]) "Cardboard and Hook Trap") return User's input return True return False return True return the value for return False ask the player what they want to horn_input do and display the game menu horn_input = cheddar = sound_horn() setup_trap() end end end end change_cheese(name, trap, cheese, e_flag) check if \angle if they \ge return the chosen trap return "Cardbboard and Hook Trap', **e_flag** if **cheddar** is **(horn_input** is choose to go name False change cheese start Complete_train() with argument of hunt_status returned from basic_hunt hunt_status = horn input = basic_hunt(cheddar[1] sound_horn() call change cheese function while True horn_input) with relevant arguments Start check if complete_train(hunt_status) return the chosen trap *horn_input* is ___True_**→** name and **e_flag** Define tuple that contains different trap names ("High end Print the available cheese list and prompt the Strain Steel Trap", "Hot Tub Trap", Training is user to type cheese name to arm the trap "Cardboard and Hook Trap") called *Traps* completed end hunt_status = / Print instructions for trap setup /basic_hunt(cheddar[1], horn_input) display training completion return False, None True if user input message Call user_input() function with a 'back' prompt that Ask user to view traps that Larry is holding by press enter and store / complete_train(hunt_status) the returned value in *view_traps* check if the user's input is a valid cheese name from the given cheese list Call user_input() with a prompt if view_traps is` return False return for whether if user want to ontinue or not, and store the Start return result in **training** print out the name of the two if the given traps that Larry is holding —False→ display 'No such cheese' Basic_hunt()with cheese name argument "cheddar" and If cheddar == 1 and check horn_input check if horn_input == "yes" check if Call user_input() function with a *training* equal ` if the user input **cheddar** is prompt that Ask the user to select a False to 'no' or cheddar is not False and trap by tying left or rightand store the set but the horn is not False and **training training** is returned value in *choice* sounded OR Trap is not set but is True $\widetilde{}$ check if the $\widetilde{}$ the horn is sounded quality of cheese False display 'Out of cheese' let the value for let the value for Print "Caught a Brown mouse! **choice_of_trap** become the **choice_of_trap** become the is positive check if first element of the trap tuple second element of the trap horn are both not set the user input "left" define at the beginging tuple define at the beginging return "Cardbboard and and sounded or "right" Display "Nothing happens. Hook Trap', **e_flag** To catch a mouse, you let the value for *hunt_status* be need both trap and True check if the cheese!" Assign the value of ✓ Print choice confirmation / cheddar name is cheddar to be 0 and the and set the trap Display "Nothing happens." trap to be 'Cardboard and True if choice is the default trap or Prompt the user to arm the trap with the selected cheese Hook Trap' let the value for **cheddar** return the value for let the value for hunt_status be become 1 hunt_status print("Invalid command! No trap selected. Larry: Odds are slim with no trap!''') return the chosen trap return False, None ___True_ end name and **e_flag** return the chosen let the value for **choice_of_trap** trap name and **e_flag** become the last element of the trap tuple define at the beginging let the value for *cheddar* become check if user print the trap is armed with Yes confirm to arm the the selected cheese trap return value for *choice_of_trap* and cheddar return True, and cheese name loop again