

Independent Paths (CheckUserInput)	Input	Output	Type
start -> input == "" ? true -> print("input cant be empty") -> raise ValueError -> end	input = ""	ValueError	visible
start -> input == "" ? false -> try -> val = int(input) -> return val -> end	input = "10"	10	visible
start -> input == "" ? false -> try -> catch ValueError -> try -> val = float(input) -> return val -> end	input = "10.1"	10.1	visible
start -> input == "" ? false -> try -> catch ValueError -> try -> Catch ValueError -> print(input is not number) -> raise ValueError -> end	input = "invalid"	ValueError	visible

Independent Paths (Calculate)	Input	Output	Type
start -> not num1 or not num2 ? true -> print("inputs can not be null") -> raise ValueError -> end	num1 = None ,num2 = None, choice = 1	ValueError	visible
start -> not num1 or not num2 ? false -> num1=check_user_input(num1) -> num2 = check_user_input(num2) -> result = "" -> choice in (1,2,3,4) ? false -> raise Exception -> end	num1 = 5, num2 = 5, choice = 5	Exception	visible
start -> not num1 or not num2 ? false -> num1= check_user_input(num1) -> num2 = check_user_input(num2) -> result = "" -> choice in (1,2,3,4) ? true -> choice == "1" ? true -> result = add(num1,num2) -> return result -> end	num1 = 5, num2 = 5, choice = 1	result = 10	visible
start -> not num1 or not num2 ? false -> num1= check_user_input(num1) -> num2 = check_user_input(num2) -> result = "" -> choice in (1,2,3,4) ? true -> choice == "1" ? false -> choice == "2" ? true -> result = subtract(num1,num2) -> return result -> end	num1 = 6, num2 = 5, choice = 2	result = 1	visible
start -> not num1 or not num2 ? false -> num1= check_user_input(num1) -> num2 = check_user_input(num2) -> result = "" -> choice in (1,2,3,4) ? true -> choice == "1" ? false -> choice == "2" ? false -> choice == "3" ? true -> result = multiply(num1,num2) -> return result -> end	num1 = 10, num2 = 20, choice = 3	result = 200	visible
start -> not num1 or not num2 ? false -> num1= check_user_input(num1) -> num2 = check_user_input(num2) -> result = "" -> choice in (1,2,3,4) ? true -> choice == "1" ? false -> choice == "2" ? false -> choice == "3" ? false -> choice == "4" ? false -> reutrn result -> end	num1 = 10, num2 = 2, choice = 4	result = 5	visible
start -> not num1 or not num2 ? false -> num1= check_user_input(num1) -> num2 = check_user_input(num2) -> result = "" -> choice in (1,2,3,4) ? true -> choice == "1" ? false -> choice == "2" ? false -> choice == "3" ? false -> choice == "4" ? true -> num2==0 ? true -> print("devide by zero") -> raise ZeroDivisionError -> end	num1 = 1, num2 = 0, choice = 4	ZeroDivisionError	Invisible
start -> not num1 or not num2 ? false -> num1= check_user_input(num1) -> num2 = check_user_input(num2) -> result = "" -> choice in (1,2,3,4) ? true -> choice == "1" ? false -> choice == "2" ? false -> choice == "3" ? false -> choice == "4" ? true -> num2==0 ? true -> print("devide by zero") -> raise ZeroDivisionError -> end	num1 = 10, num2 = '0', choice = 4	ZeroDivisionError	visible

Independent Paths (Add)	Input	Output	Type
start -> result = x+y -> return result -> end	x = 5 , y = 6	result = 11	visible

Independent Paths (Subtract)	Input	Output	Type
start -> result=x-y -> return result -> end	x = 15 , y = 10	result = 5	visible

Independent Paths (Multiply)	Input	Output	Type
start -> result=x*y -> return result -> end	x = 20 , y = 10	result = 200	visible

Independent Paths (Divide)	Input	Output	Type
start -> y==0 ? true -> print("you can't divide by zero!") -> raise ZeroDivisionError -> end	x = 5 , y = 0	ZeroDivisionError	visible
start -> y==0 ? false -> x==0 ? true -> return 0 -> end	x = 0 , y = 5	0	visible
start -> y==0 ? false -> x==0 ? false -> result= x/y -> return result -> end	x = 15 , y = 5	result = 3	visible

Independent Paths (isExit)	Input	Output	Type
start -> next_calculation == "no" ? true -> return true -> end	next_calculation = "no"	TRUE	visible
start -> next_calculation == "no" (false) -> next_calculation == "yes" (true) -> return false -> end	next_calculation = "yes"	FALSE	visible
start -> next_calculation == "no" ? false -> next_calculation == "yes" ? false -> raise ValueError -> end	next_calculation = "invalid"	ValueError	visible