Creating the solution

The code starts with the main program. I have added comments in the program to explain the processing. I have a main loop that continues until the user wants to exit by typing Exit. Inside that loop a user input is validated and checked by calling other functions.

CowsandBullsFinal.py - C:\Users\Gemma\Documents\NEW GCSE Comp Sci\NEA\CowsandBullsFinal.py

File Edit Format Run Options Windows Help

```
#-------------------------#
# Main program            #
#-------------------------#

#Introduction to the game
print("****************************************************")
print("* Welcome to the AQA Bulls and Cows program.      *")
print("* A random 4 digit number has been generated.     *")
print("* A number will not be repeated in the 4 digits   *")
print("* Please enter your first guess at the number.    *")
print("* You will be told how many numbers were right    *")
print("* A bull = both number and position is correct    *")
print("* A cow = correct number but wrong position       *")
print("****************************************************")
print("")

#calling the function that generates the number
gen_number = generate_number()
print(gen_number) # Just for testing.  Comment this line out when playing for re

#sets a data variable for user input
user_input = ""
#an integer is created to keep count of the number of guesses
guesses = 0

#loop that allows a number to be entered until the program is exited
#or the number is guessed
while user_input != "exit":
    print("Please enter a 4 digit number or exit")
    user_input = raw_input("number: ")

    #breaks the loop if exit is entered, the game ends
    if user_input == "exit":
        break

    #checks to see if the number has been guessed correctly
    #the validation function also checks that a valid number has
    #been enterd
    if validate_number(user_input) is True:
        guesses += 1
        if check_guess(user_input, gen_number) is True:
            print ("Congratulations you have guessed the number in " + str(guess
            break

#end of while loop

print ("goodbye")
```

Ln: 138 Col: 17

Before the main loop there is a call to a function called **generate_number().** This function returns a list data structure to the main program and is saved in the variable **gen_number**.

```
File  Edit  Format  Run  Options  Windows  Help
#-------------------------#
# AQA Cows and Bulls Game #
# Written in Python 3.3   #
# January 2016            #
# Candidate: Harry Potter #
# Centre: Hogwarts        #
# Centre No: 000000       #
# Candidate No: 9999      #
#-------------------------#

#import libraries
import random #imports random for random number

#Function to generate a random 4 digit number
def generate_number():
    #Creates a list to be used to guess the number
    #loop to generate 4 numbers. Continues until all numbers are unique
    gen_number = []
    i = 0
    while i <=3:
        random_value = random.randrange(0, 9)
        #IF statement to check that the number has not been used before
        if gen_number.count(random_value) == 0:
            gen_number.append(random_value)
            i+=1
    return gen_number
                                                                Ln: 108 Col: 0
```

This function uses the python count method to count the occurrence of a value in a list and the append method to add a value to a list. It also used the **random.randrange** function, which required a python library to be imported.

The main program then checks that a user entered number is valid by calling a function called **validate_number**. This function is passed the **user_input** and returns a Boolean value indicating if the user input has passed the validation and so can be checked against the randomly generated number **gen_number**. A loop is used to look at each digit in the user input.

The **user_input** variable has been passed by the main program. It was manually entered using the **raw_input()** function. Originally I used the input() function but python creates a variable that takes the form of the data entered when this function is used. Therefore the user_input would be created as an integer if a 4 digit number is entered, but this is no use for this program as a string is better because it can be used as a list and looped through. I originally used this and got an error, so changed to use raw_input() function that always returns a string.

```
File  Edit  Format  Run  Options  Windows  Help
def validate_number(user_input):
    #validation to check that the user entry is a number and 4 digits in length
    #the user_input is also checked for duplicate numbers. If validation fails tl
    # the function returns False.
    test_input = ""

    #try to convert the user input to an integer.  If there are characters
    #in the input then a value error is returned.
    try:
        test_input = int(user_input)
    except ValueError:
        print("you must enter a number")
        return False

    #the length of the number must be 4 digits long.  The format of the input
    #has already been checked so it is a number
    if len(user_input) != 4:
        print("your number should be exactly 4 digits long")
        return False
    i = 0
    while i <=3:
        #IF statement to check that the digit has not been used before
        #loop through each digit in the input string and count the number
        #of times it occurs which should only be once, otherwise it is a
        #duplicate
        if user_input.count(user_input[i]) != 1:
            print("your number contains duplicates")
            return False
        i+=1
    # end loop
    return True
                                                                    Ln: 133 Col: 0
```

The final part of the main program calls the **function check_guess**. This function is passed the **user_input** and **gen_number** variables. Because these two parameters can be interpreted as lists they will be compared item by item in order to determine if the match makes a bull or a cow.

The **check_guess** procedure returns True if there are 4 bulls, meaning the number has been guessed correctly, or False if it hasn't. A count of the number of guesses is kept in the main program.

The main program finally ends when exit has been entered by the user OR the number has been guessed correctly.

There is a full code listing in the appendix.

File Edit Format Run Options Windows Help

```
def check_guess(user_input, gen_number):
    # initialise all the variables to count bulls and cows
    i = 0
    bulls = 0
    cows = 0

    #a loop that looks at each digit in the user input and compares it to the ge
    #number.

    while i <=3:
        #if the input digit is the same as the generated digit in the same posit
        #then it is a bull

        if gen_number[i] == int(user_input[i]):
            bulls = bulls + 1

        #if the digit is in the generated number but in a different position
        #then it is a cow
        elif int(user_input[i]) in gen_number:
            cows = cows + 1
        i += 1
    # end of loop

    print ("You have " + str(bulls) + " bulls and " + str(cows) + " cows")

    #If there are 4 bulls then True is returned to the main program
    #so that a congratulations message can be printed
    if bulls == 4:
        return True
    else:
        return False
```

Ln: 87 Col: 18