Computer Science I
Project 1: Animation

TAERAKUL Janat, #19B60096

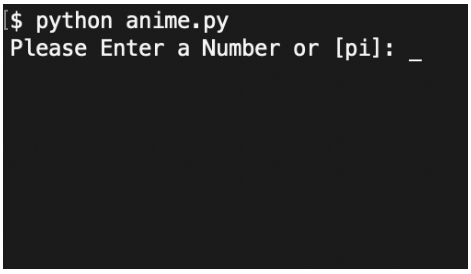
November 9, 2019

1 Instruction

This program is a python program. It takes a number as an input and shows the number on a sliding 10 × 28 pixels display. The display consists of numbers 0 and 1, which denote black and white pixels. Another function of this program is displaying first 100 digits of Pi (π).

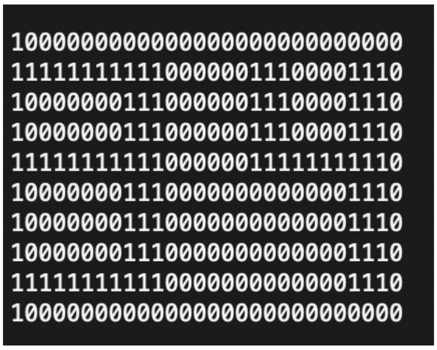
To run this program, open terminal and go to the directory that contain `anime.py` file, then run `$ python anime.py`.

After the program is run, it will ask its user to enter a number or ”pi” as shown in figure (1). If the user enter a number, the program will show that number moving right to left until the end of that number as figure (2). If the user enter ”pi”, the program will shows the first 100 digits of Pi (π). In case the user enter something other than numbers and dot, the program will ignore that character and skip to the next one.



```
[$ python anime.py
Please Enter a Number or [pi]: _
```

Figure 1: The program ask user to input a number.



```
1000000000000000000000000000
1111111111100000011100001110
1000000011100000011100001110
1000000011100000011100001110
1111111111100000011111111110
1000000011100000000000001110
1000000011100000000000001110
1000000011100000000000001110
1111111111100000000000001110
1000000000000000000000000000
```

Figure 2: The program showing the number inputed by user.

2 How the Program Works

To do this project, I did research on how to define functions in python3. This program consists of several variables and functions which are explained as following.

2.1 Global Variables

1. `img` - An array of 10 integers. Used as display of the program.
2. `num` - A 2D array with 10 arrays of 10 integers. Used as templates for number 0-9.
3. `dot` - An array of 10 integers. Used as template of a dot.
4. `pi` - A string of first 100 digits of Pi (π)

2.2 Function

1. **tenPow(n)** - Input an integer n into this function,it returns 10^n back.

   The variable *ans* initially has value of 1. The function has a loop for n times, each time it multiply *ans* by 10 and return *ans* when the loop ends.

   ```
   def tenPow(n):
     ans = 1
     for i in range(n):
       ans *= 10
     return ans
   ```

2. **mod(a, b)** - Input integer a and b into this function, it returns the remainder from dividing a by b. I need modulo function to find the last digit of a number (Ex. $1324 \ mod\ 10 = 4$).

   The function can be implemented by subtracting $(a//b) * b$ from a. Since // is integer division, $(a//b) * b$ gives value of the most multiple of b that less than or equal to a.

   ```
   def mod(a, b):
     return a - ((a//b) * b)
   ```

3. **kthDigit(x, k)** - Input integer x and k into this function, it returns the k^{th} digit from the last of x.

   The function uses $mod 10$ to find the last digit of $x//10^{k-1}$, which is k^{th}, $(k+1)^{th}$, ... digits of x. (Example. $1234567//10^{3-1} = 12345$ and $12345 \ mod\ 10 = 5$ which is the 3^{rd} digit of x)

   ```
   def kthDigit(x, k):
     return mod(x // tenPow(k-1), 10)
   ```

4. **printImg()** - This function prints the value in variable *img* and call function `time.sleep(0.1)` to pause the program for 0.1 second.

```
def printImg():
  for line in img:
    print(line)
  print()
  time.sleep(0.1)
```

5. **showDigit(x)** - Input a number x (0-9) into this function, it push the template in variable *num* of the number x into the display or the variable *img*.

   The function consists of two nested loops. The outer loop repeats 16 times as $k = 16, 15, ..., 2, 1$. The inner loop repeats 10 times as $i = 0, 1, ..., 8, 9$, each time it push the k^{th} digit of i^{th} integer in the template of number x into the i^{th} integer of variable img.

   To shift all digits of $img[i]$ to the left, the function multiplies $img[i]$ by 10. Then add the k^{th} digit of each integer in the template of number x by using function `kthDigit()`. Now, $img[i]$ has 29 digits, which has to be changed to 28 digits. So, the function subtracts 10^{28} from $img[i]$. Lastly, if $img[i]$ does not have the 28^{th} digit from the last then the function adds 10^{27} to $img[i]$.

```
def showDigit(x):
  for k in range(16, 0, -1):
    for i in range(10):
      img[i] *= 10
      img[i] += kthDigit(num[x][i], k)
      img[i] -= tenPow(28)
      img[i] += (1 - kthDigit(img[i],28)) * tenPow(27)
    printImg()
```

6. **showDot()** - This function push the template of a dot in the variable *dot* into the display or the variable *img* uy using the same method as function `addDigit()`.

```
def showDot():
  for k in range(10, 0, -1):
    for i in range(10):
      img[i] *= 10
      img[i] += kthDigit(dot[i], k)
      img[i] -= tenPow(28)
      img[i] += (1 - kthDigit(img[i],28)) * tenPow(27)
    printImg()
```

7. **showBlank()** - This function add 28 columns of zero into the display or the variable *img* uy using the same method as function `addDigit()` and function `addDot()`.

```
def showBlank():
  for k in range(28):
    for i in range(10):
      img[i] *= 10
      img[i] -= tenPow(28)
      img[i] += (1 - kthDigit(img[i],28)) * tenPow(27)
    printImg()
```

8. **show(ch)** - Input a character into this function, it decides to call a function between function `addDigit()` and function `addDot()`. If the input is a number, it calls function `addDigit()` and pass the input into the function. If the input is a dot, function `addDot()`. The function will ignore other types of input.

```
def show(ch):
  if ch == '.':
    showDot()
  elif ch >= 48 and ch <= 57:
    showDigit(ch - 48)
```

9. **showNum(str)** - Input a string of a number into this function, it decodes the string to a list of characters and pass each character into function `show()`. After it shows all character, it fills the display (variable *img*) with zeros.

```
def printNum(s):
  ch = s.encode("ascii")
  for digit in ch:
    show(digit)
  showBlank()
```

2.3 Main Program

When the program is run, the process starts from asking for input string from user. If the input string match ”PI”, ”Pi”, ”pI”, or ”pi”, the program sets value of input variable to pi (string of ”3.14159...”). Then, the program passes the input into function `printNum()`

```
inp = input("Please Enter a Number or [pi]: ")
if inp == 'PI' or inp == 'pi' or inp == 'Pi' or inp == 'pI':
  inp = pi
printNum(inp)
```