AP Computer Science Principles | Test 1 | Name: _____
Fall 2017 | Mr. Alwin Tareen

Part I. (47 points) Solve each of the following problems. For the multiple choice problems, select the correct answer by placing an "X" in the box beside it.

(1pt) **1.** Which of the following choices is a legal and legitimate `Python` variable name?

- ☐ `2bad4you`
- ☐ `calvin&hobbes`
- ☐ `year2000`
- ☐ `#hammertime`

1 pt

(1pt) **2.** You would like to set up a variable called `ounces` that has the value `16`. What simple `Python` statement will accomplish this?

- ☐ `ounces = 16`
- ☐ `16 = ounces`
- ☐ `def ounces(16):`
- ☐ `ounces(16)`

1 pt

(1pt) **3.** What does the following `Python` statement print out:

```
print("123" + "abc")
```

- ☐ `"123" + "abc"`
- ☐ This is a syntax error because you cannot add strings.
- ☐ `123+abc`
- ☐ `123abc`

1 pt

(1pt) **4.** In `Python`, the `float` data type is used to store:

- ☐ booleans
- ☐ decimal numbers
- ☐ strings
- ☐ integers

1 pt

(1pt) **5.** What is the result of the following `Python` statement:

```
print(42%10)
```

- ☐ `1042`
- ☐ `420`
- ☐ `4`
- ☐ `2`

1 pt

(1pt) **6.** Which of the following choices is the correct assignment statement for a `string` data type?

- ☐ `greetings = [Hello]`
- ☐ `greetings = @Hello@`
- ☐ `greetings = "Hello"`
- ☐ `greetings = #Hello#`

1 pt

(1pt) **7.** What is the result of the following `Python` statement:

```
print(17/4)
```

- ☐ `4`
- ☐ `4.0`
- ☐ `4.3`
- ☐ `4.25`

1 pt

7 pts

(1pt) **8.** What are the only values that are permissible in `Python`'s `boolean` data type?

- ☐ `Yes, No`
- ☐ `On, Off`
- ☐ `Right, Wrong`
- ☐ `True, False`

1 pt

(1pt) **9.** Which of the following is a comment in `Python`?

- ☐ `/* This is a test */`
- ☐ `// This is a test`
- ☐ `# This is a test`
- ☐ `% This is a test`

1 pt

(1pt) **10.** Which of the following elements of a mathematical expression in `Python` is evaluated first?

- ☐ Multiplication `*`
- ☐ Addition `+`
- ☐ Parenthesis `()`
- ☐ Subtraction `-`

1 pt

(1pt) **11.** What will be the value of `x` when the following statement is executed: `x = int(98.6)`

- ☐ `99`
- ☐ `6`
- ☐ `98`
- ☐ `100`

1 pt

(1pt) **12.** What does the `Python` function `input()` do?

- ☐ Pause the program and read data from the user.
- ☐ Take a screen shot from an area of the screen.
- ☐ Read the memory of the running program.
- ☐ Connect to the network and retrieve a web page.

1 pt

(1pt) **13.** Which `Python` keyword indicates the start of a function definition?

- ☐ `sweet`
- ☐ `def`
- ☐ `continue`
- ☐ `return`

1 pt

(1pt) **14.** Consider the following function definition:

```
def circlearea(radius):
```

In this context, what is the formal name for the variable `radius`?

- ☐ expression
- ☐ logical deduction
- ☐ parameter
- ☐ condition

1 pt

(1pt) **15.** Which of the following is NOT a valid string method in `Python`?

- ☐ `boldface()`
- ☐ `startswith()`
- ☐ `upper()`
- ☐ `strip()`

1 pt

8 pts

(1pt) **16.** What does the following Python program print out?

```
str1 = "Hello"
str2 = "there"
greet = str1 + str2
print(greet)
```

- ☐ `Hello there`
- ☐ `Hellothere`
- ☐ `there`
- ☐ `Hello`

1 pt

(1pt) **17.** How would you use the index operator to print out the letter `"q"` from the following string?

```
x = "From marquard@uct.ac.za"
```

- ☐ `print(x[9])`
- ☐ `print(x[8])`
- ☐ `print(x[-1])`
- ☐ `print(x[q])`

1 pt

(1pt) **18.** How would you use string slicing to print out `"uct"` from the following string?

```
x = "From marquard@uct.ac.za"
```

- ☐ `print(x[14+17])`
- ☐ `print(x[15:18])`
- ☐ `print(x[14:17])`
- ☐ `print(x[14:3])`

1 pt

(1pt) **19.** What is the iteration variable in the following `Python` code?

```
for letter in "banana":
    print(letter)
```

- ☐ `letter`
- ☐ `print`
- ☐ `in`
- ☐ `"banana"`

1 pt

(1pt) **20.** How would you print out the following string in all upper case in `Python`?

```
greet = "Hello there"
```

- ☐ `puts greet.ucase;`
- ☐ `print(uc($greet))`
- ☐ `print(greet.upper())`
- ☐ `console.log(greet.toUpperCase());`

1 pt

(1pt) **21.** What does the following Python program print out?

```
data = "From stephen.marquard@uct.ac.za"
pos = data.find(".")
print(data[pos:pos+3])
```

- ☐ `uct`
- ☐ `mar`
- ☐ `.ma`
- ☐ `ste`

1 pt

6 pts

(1pt) **22.** What is the output of the following Python code:

```
print(3 > 4 or (2 < 3 and 9 > 10))
```

Answer:

1 pt

(1pt) **23.** What is the output of the following Python code:

```
def choose(x, y, z):
    if x:
        return y
    else:
        return z
print(choose(False, 2, 3))
```

Answer:

1 pt

(1pt) **24.** What is the output of the following Python code:

```
lunch = "cheeseburgers"
print(lunch[6:12])
```

Answer:

1 pt

(1pt) **25.** What is the output of the following Python code:

```
greeting = "Hello, world!"
newgreeting = "J" + greeting[1:]
print(newgreeting)
```

Answer:

1 pt

(1pt) **26.** What is the output of the following Python code:

```
print("cola" in "chocolate")
```

Answer:

1 pt

(1pt) **27.** What is the output of the following Python code:

```
print("seed" in "banana")
```

Answer:

1 pt

(1pt) **28.** What is the output of the following Python code:

```
fruit = "strawberry"
bigfruit = fruit.upper()
print(bigfruit)
```

Answer:

1 pt

(1pt) **29.** What is the output of the following Python code:

```
vegetable = "cauliflower"
index = vegetable.find("u")
print(index)
```

Answer:

1 pt

(1pt) **30.** What is the output of the following Python code:

```
line = "Please have a nice day"
print(line.startswith("Please"))
```

Answer:

1 pt

(1pt) **31.** What is the output of the following Python code:

```
meal = "fresh pizza is the best pizza"
print(meal.replace("pizza", "salad"))
```

Answer:

1 pt

10 pts

(1pt) **32.** What is the output of the following `while` loop?

```
num = 0
while num < 3:
    print(num)
    num += 1
```

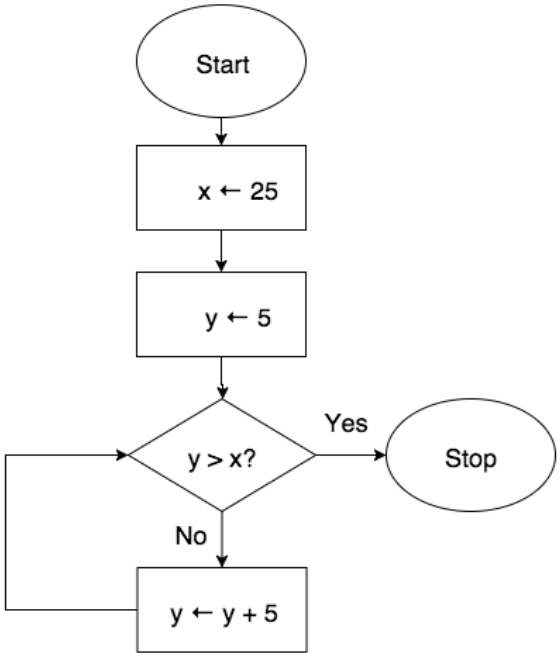
1 pt

(1pt) **33.** What is the output of the following `while` loop?

```
num = 5
while num < 10:
    print(num)
    num += 2
```

1 pt

(2pts) **34.** Write a piece of `Python` code that implements the functionality of the following flowchart:



2 pts

4 pts

(2pts) **35.** Assume that `sample` is a string of lower case text characters. Write a `Python` function that counts the number of vowels that are contained in the string `sample`. Valid vowels are: `"a", "e", "i", "o", "u"`. Your function should be called `countvowels(sample)`, which takes in a single parameter, `sample`. The function should return an integer which is the total quantity of vowels in the string.

2 pts

- The output of your program should be `5` if the following statements are executed:

```
result = countvowels("azcbobobegghakl")
print(result)
```

(2pts) **36.** Write a `Python` function that takes in a string as a parameter, and generates a new string, which is made up of three copies of the last two characters of the original string. Your function should be called `extraend(word)`, which takes in a single parameter, `word`. The function should return a string.

2 pts

- The output of your program should be `lololo` if the following statements are executed:

```
result = extraend("hello")
print(result)
```

4 pts

(4pts) **37.** Pig Latin is a type of slang language that is easy to learn and understand. An English word can be translated into Pig Latin by following these two simple rules:

4 pts

- If the English word begins with a vowel, then the corresponding Pig Latin word is generated by appending the letters `"hay"` to the end of the word. For example, `"orange"` becomes `"orangehay"`.
- If the English word begins with a consonant, then the corresponding Pig Latin word is generated by moving the first letter to the end of the word, then appending the letters `"ay"`. For example, `"peach"` becomes `"eachpay"`.

Write a `Python` function that takes in an English word as a parameter, and translates that word to Pig Latin. Your function should be called `piglatin(word)`, which takes in a single parameter, `word`. The function should return a string which is the Pig Latin translation of `word`.

- The output of your program should be `orangehay` if the following statements are executed:

```
result = piglatin("orange")
print(result)
```

- The output of your program should be `eachpay` if the following statements are executed:

```
result = piglatin("peach")
print(result)
```

4 pts

(4pts) **38.** In this question, you will write a `Python` function that implements an encryption technique known as the Caesar Cipher. The key idea behind the Caesar Cipher is to shift each letter in a secret message by a fixed number of positions. If this shifting behaviour goes further than the end of the alphabet, then it "wraps around" to the beginning, and continues from there.

4 pts

`Python` uses a numerical `ASCII` value to represent each text character in the alphabet. For example, the `ASCII` value of the letter `"a"` is `97`, and the `ASCII` value of the letter `"z"` is `122`. In order to convert between the two formats, you need to use the following `Python` functions:

`ord(txt)` This returns the numerical `ASCII` code corresponding to the text character `txt`.

`chr(num)` This returns the text character corresponding to the numerical `ASCII` code `num`.

Unencrypted text is generally called *plaintext*, and encrypted text is generally known as *ciphertext*.

In general, the Caesar Cipher encrypts messages by "rotating" each letter by k positions. More formally, if p is the `ASCII` value of a letter in the plaintext, and k is the amount by which each letter is shifted, then the `ASCII` value of the corresponding letter in the ciphertext c, is computed by the following equation:

$$c \quad = \quad (p + k) \bmod 26$$

Write a `Python` function called `caesarcipher(plaintext, shift)` that takes in a `plaintext` message and `shift` value as parameters, and returns the encrypted version of this message by using the Caesar Cipher. Assume that the `plaintext` message only consists of lowercase text characters.

- The output of your program should be `qechec` if the following statements are executed:

```
result = caesarcipher("mayday", 4)
print(result)
```

4 pts