

CSE 160 Section 2 Problems

1. Fill in the necessary code to build the list ages

```
ages = [] #creates an empty list
for age in [20, 21, 20, 22, 19, 18, 14, 35]:

    print ages
```

The output should be [20, 21, 20, 22, 19, 18, 14, 35]
(Hint: x.append(2) appends the number 2 to the end of list x)

2. Write a function, over_twenty(ages) to count the number of people over 20 years old in the list ages.

3. Write the output to the following problem:

```
grid = [[1, 2, 3], ['a', 'b', 'c'], ['c', 's', 'e'], [1, 6, 0]]
print grid[0][0]
print grid[1][2]
print grid[2][1]
print grid[3][2]
```

4. Modify the following code so that it properly adds 5 to everyone's age

```
ages = [20, 21, 20, 22, 19, 18, 14, 35]
for i in ages:
    ages[i] + 5
print ages
```

print ages should now return [25, 26, 25, 27, 24, 23, 19, 40]

5. Write a function that calculates and returns the average of ages. You are not allowed to use python's built-in `sum()` function. Your function should take in the list `ages` as a parameter and return the average.

6. Given a function `get_height` that computes the height of the student passed in, write a new function `max_height` that finds the maximum height of all the people in the class. Your function should take in a list of student names and return the maximum height. You can assume height is in inches and that the list of all students in the class is `class_lst`.

`get_height('nicholas')` will return 75

What is the type of `max_height(students)`?

Suppose the code was modified to `print max_height` instead of `return max_height`, what would be the type of `max_height(students)`?

CSE 160 Section 02 Solutions

1.

```
ages = [] #creates an empty list
for age in [20, 21, 20, 22, 19, 18, 14, 35]:
    ages.append(age)
print ages
```

2.

```
def over_twenty(ages):
    total = 0
    for age in ages:
        if age > 20:
            total = total + 1
    return total
```

3.

```
1
c
s
0
```

4.

```
ages = [20, 21, 20, 22, 19, 18, 14, 35]
for i in range(len(ages)):
    ages[i] = ages[i] + 5
print ages
```

5.

```
def avg_age(ages):
    total = 0
    for age in ages:
        total = total + age
    avg = float(total) / len(ages)
    return avg
```

6.

```
def max_height(class_lst):
    cur_max = 0
    for student in class_lst:
        student_height = get_height(student)
        if(student_height > cur_max):
            cur_max = student_height
    return cur_max
```

Type when returning: Int

Type when printing: None

CSE 160 Section 02 Code Examples

1. An example of opening a file and printing all of its contents:

```
input = open("input.txt", "r")
for line in input:
    print line,

input.close()
```

The comma at the end of the print statement stops python from printing a redundant `\n` in addition to the one that is already in the line variable.

2. An example of opening a file and counting the amount of lines it contains:

```
input = open("input.txt", "r")
line_number = 0
for line in input:
    line_number = line_number + 1

input.close()

print line_number
```

3. An example of opening a file and only printing the odd numbered lines (where the first line is 1):

```
input = open("input.txt", "r")
line_number = 0
for line in input:
    if line_number % 2 == 0:
        print line,
    line_number = line_number + 1

input.close()
```

4. An example of opening a file and writing to it:

```
output_file = open("output.txt", "w")

output_file.write("apple\n")
output_file.write("banana\n")
output_file.write("nyan\n")

output_file.close()
```

5. An example of some basic uses of a list:

```
data = []
data.append(39)
data.append(9001)
data.append(42)
# data = [39, 9001, 42] at this point

data.sort()
# data = [39, 42, 9001] at this point

# Report the amount of entries in the list, which is 3
print len(data)

# Prints each value in the list
for entry in data:
    print entry

# An alternative way to print each value in the list
for i in range(len(data)):
    print data[i]
```

6. An example of a function definition:

```
# Returns the sum of the three given arguments
def example_function(a, b, c):
    return a + b + c
```

7. An example of `rstrip()`:

```
example_string = "A string with a newline at the end\n"
stripped_string = example_string.rstrip()

# The contents of the two variables:
# example_string = "A string with a newline at the end\n"
# stripped_string = "A string with a newline at the end"
```

8. An example of String formatting:

```
a = "%s => %d, %f" % ("Example String", 42, 9001.0)
print a # Example String => 42, 9001
```