Do not open this exam until you are told. Read these instructions:

1. This is a closed book exam. No calculators, notes, or other aids are allowed. If you have a question during the exam, please come to the front of the class.
2. You must turn in your exam immediately when time is called at the end.
3. The exams adds up to 45 points total. 120 minutes. Each question’s point value is indicated.
4. In order to be eligible for as much partial credit as possible, show all of your work for each problem, write legibly, and clearly indicate your answers. Credit cannot be given for illegible answers.

7. Fill in the following:

NAME :

G# :

(write your name and G# on your scantron also)

DO NOT OPEN UNTIL TOLD TO START
But ask me about my hint for returns!
A. Trace through the following code using the scratch space below, and then write your answer in the box on the right-hand side. Then, TRANSFER YOUR ANSWER to the scantron sheet line-by-line. print ERROR if the line would have raised an exception. (10 points)

```python
def func1(x,y,lis):
    print lis[y:x]
    lis[x] = lis[y]
    lis = [x]
    return lis

def func2(a,b,nums):
    print a in nums.keys()
    print nums[b]
    nums[1] = 2

def main(x):
    x = x + 1
    lis = [4,6,8,9,10]
    things = [7,lis,x]
    lis = func1(x,2,lis)
    print lis
    print things
    if x == 3:
        dict = {x:x,1:lis,"c":"dog"}
        print func2('x',0,dict)
        print dict[1]

main(2)
main(1)
```

WILL NOT BE GRADED!
COPY TO SCANTRON!

A1_____________________________________________
A2_____________________________________________
A3_____________________________________________
A4_____________________________________________
A5_____________________________________________
A6_____________________________________________
A7_____________________________________________
A8_____________________________________________
A9_____________________________________________
A10_____________________________________________
B. Trace through the following code using the scratch space below, and then write your answer in the box on the right-hand side. Then, TRANSFER YOUR ANSWER to the scantron sheet line-by-line. (9 points)

```python
def func1(x, lis):
    print("lis[x]: " + lis[x])
    if x > 0:
        x -= 1
    else:
        x += 6
    try:
        integer = int(lis[x])
        print("div: " + str(5//integer))
        print("all good")
    except TypeError:
        print("incompatible types!")
    except ZeroDivisionError:
        print("div by zero!")
    raise ValueError()
    else:
        print("func1")
    return "done func1"

def main(x):
    try:
        print(func1(x, ["c", 2, "1", "0", "7"]))
        print("success")
    except TypeError:
        print("wrong")
    except Exception:
        print("oops")

main(2)
main(0)
main(1)
main(4)
```

WILL NOT BE GRADED!
COPY TO SCANTRON!

<table>
<thead>
<tr>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
<th>B5</th>
<th>B6</th>
<th>B7</th>
<th>B8</th>
<th>B9</th>
<th>B10</th>
<th>B11</th>
<th>B12</th>
</tr>
</thead>
</table>
Multiple choice.

1. What is the correct method signature, defined in person.py, of the constructor for a class called Person, which can be called as

   ```python
   p = Person('Sally',23) ?
   ```

   a. `def Person(self, name, age):
      
   b. `def __init__(self, name, age):
      
   c. `def __init__(self, name, age, zip):
      
   d. a and b
   
   e. b and c

2. Imagine I wrote a Person class with a constructor that sets an attribute called name to the value of an incoming argument. What is wrong with the following method of the same class, defined in the appropriate place in the same module, that wants to set the name attribute to another incoming argument?

   ```python
   def foo(self, newName):
       name = newName
   ```

   a. name cannot be reset after it was initialized
   
   b. foo can be called without ever creating an object
   
   c. The method has no effect on the attribute name of the object
   
   d. Nothing; what is written above is correct

3. If the constructor for a class called Person can only be called with two arguments (no less, no more), what must be true about the attributes of a Person object?

   a. A Person object can only have at most two attributes
   
   b. A Person object can have more than two attributes
   
   c. The Person constructor takes a third argument with a default value

4. The statement `from project2 import *` imports all the functions in the file project2.py into the file where this statement was written

   a. True   
   b. False

5. In a try-except block, the finally clause is not executed if an exception was raised in the matching try block.

   a. True   
   b. False

6. If you try to open a file for reading in python, and it doesn't exist, you will receive an error message

   a. True   
   b. False

7. If `lis1 = [4, 7, 1]` and `lis2 = [8,2,0]`, what does `lis2[lis1[2]]` evaluate to?

   a. 1   
   b. 4   
   c. 2   
   d. ERROR
8. \( \text{range(len([8,9]))} \) simplifies to which of the following?

a. [ ]

b. [0, 1]

c. [1, 2]

d. ERROR

\[ \text{CHOOSE ONE} \text{ of the following two problems to answer.} \]

You DO NOT have to answer both; pick the one you find more to your liking. If you attempt both problems, I WILL ONLY GRADE ONE OF THEM. If you attempt both problems, I will grade the first one.

If you want to initially attempt both problems, make it clear to me which one you want to be counted, otherwise I will only grade the first one.
Option 1:

Write a function called `scan` that takes a string as an argument and prints out one, and only one, of the following according to the argument:

- **ME** if the string `me` is in the input at least once
- **OM** if the string `om` is in the input at least once
- **MEH** if the string `meh` is in the input at least once
- **OVERLAP** if more than one of the previous three bullets are true
- **NONE** if none of the previous bullets are true

You may NOT use any built-in functions or methods besides `len`, and you may NOT use the `in` keyword.
Option 2:

Write a function called `sumEven` that accepts a list of strings as its argument. Each string will be either an integer, or the @ character. The function will return a list of sums of even integers between @ signs. It will ignore every third element in the list, regardless of that element’s contents. The incoming list is guaranteed to have an even number before an @ sign.

For example,

- `sumEven(['2'])` would return the list [2]
- `sumEven(['1'])` would return the list [0]
- `sumEven(['2','@','5'])` would return the list [2, 0]
- `sumEven(['2','3','@','4','1','6','@','3','2','@','2','@','1','2'])` would return the list [6, 0, 4]

You may NOT use any built-in functions or methods besides `int`, `len` and `append`. 
Scratch paper – rip off and turn in
Name: ____________________________