Eastern
Mediterranean
University

Name::__________________________ ____

Signature::___________________ **Group No:** ________ **Student No::**____________________

CMPE- AING -CMSE 107
Foundations of Computer-Software Engineering

Fall-2024/25 (14-11-2024) Midterm Exam

Part A

__________(28pts)

Part B

__________(24pts)

PartC

__________(27pts)

PartD

__________(30pts)

Total

(109pts)

Cem Ergün, Group 1 and 2
Zeki Bayram 3
Emre Özen, Group 4,5

Total six pages (including the cover page), four parts, 95 minutes.

- Answers that are not written into the given boxes WILL NOT BE GRADED.

- Student who ATTEMPT TO CHEAT will GET ZERO for the exam, and may be directed to disciplinary investigation for further punishment. Sharing pen, pencil, eraser, and sharpeners is considered as attempt of cheating.

- Remove all notes, books and unnecessary objects from your desk. Keep only THIS BOOKLET, PEN-PENCIL-ERASES AND YOUR ID CARD on your desk.

- Having any kind of *electronic calculators, computers, phones and gadgets such as earphones, intelligent watches* etc. at any easy accessible place is strictly not allowed. You are allowed to keep *electronics gadgets* in your bags after turning their power off. Do not keep *electronic watches* on your wrist, desk, or in your pocket.

- Talking, making any kind of noise, asking questions are not allowed. Do not talk, and do not create any sound once the exam is started.

Course Student Outcome relations:

Questions in PartA -Identify the difference between computer hardware and computer software (CO1)
Questions in PartB -Construct an algorithm for solving a computational problem (CO2 CO3)
Questions in PartD -Write a complete Python program for solving a problem (CO4)
Questions in PartC -Use of selection and repetition structures within a Python Program (CO5)

PART A: (Each is 2pt)
MULTIPLE CHOICE (**Mark only one choice**)

QA1) What does GPU stand for?

[ ] General Processing Unit
[X] Graphics Processing Unit
[ ] General Purpose Unit
[ ] Graphics Performance Unit

QA2) Which of the following is responsible for the visual output of a computer?

[ ] Hard Drive
[ ] CPU
[X] Graphics Card
[ ] RAM

QA3) What is the main purpose of an operating system?

[ ] To provide security software for the computer
[X] To serve as an interface between the user and hardware
[ ] To connect the computer to the internet
[ ] To maintain system backups

QA4) Which of the following is an example of a high-level programming language?

[ ] Assembly
[X] Python
[ ] Machine code
[ ] Binary

QA5) Which part of a computer is primarily responsible for performing arithmetic and logic operations?

[ ] Motherboard
[X] CPU
[ ] Hard Drive
[ ] RAM

QA6) What type of memory is used for temporary storage while a computer is running?

[ ] SSD
[ ] ROM
[X] RAM
[ ] Hard Drive

QA7) What is the binary representation of the decimal number 255?

[X] 11111111
[ ] 01111111
[ ] 10000001
[ ] 100000001

QA8) Which of the following best describes a compiler?

[ ] It translates machine code to source code
[X] It converts high-level language code into machine code
[ ] It directly executes source code line by line
[ ] It translates binary data to text

QA9) What is a common use for an interpreter in programming?

[ ] To compile source code into object code
[X] To debug and execute source code line by line
[ ] To manage the computer's hardware
[ ] To handle memory allocation

QA10) The idea of a brute force approach can be described as
[X] trying all possible solutions to a given problem
[ ] a computational approach that uses a great deal of memory
[ ] the most efficient, effective, and direct means to solve a problem
[ ] uses algorithmic approach for the best solution

QA11) Which of the following is False?

[ ] All algorithms must terminate after a finite amount of time
[ ] Algorithms are general computational methods for solving specific problems
[ ] The computation that a given computer performs is only as good as the underlying algorithm used.
[X] Algorithms must be written in a specific programming language to be valid

QA12) What is the final step in the problem-solving process?

[ ] Identifying the problem
[ ] Generating possible solutions
[X] **Testing t**he solutions
[ ] Implementing the solution

QA13) Which can be a correct Identifier in Python?

[ ] 0abc
[ ] -1A2a
[ ] +pythOn
[X] x_y1

QA14) An error made in the design of a solution's algorithm is a?
[ ] syntax error
[X] semantic error
[ ] computational error
[ ] unlogical error

PART B: FILL IN BLANKS QUESTIONS (Each is 3pts)

Write answers only into the answering box.

What is the output of the following piece of code?

QB1)

```
a=2
b=3
print(b*2**3/a)
```

Answer ???: 12

QB2)

```
print(3%2//4<3*2%4)
```

Answer ???: True

QB3)

```
x=False
a=1
b=3
print(b>a or x )
```

Answer ???: True

QB4)

```
x="abcdef"
y=x[1:2]+x[-1]
print(y)
```

Answer ???: bf

QB5)

```
count = 3
while count <= 10:
  count+=4-count%2
print(count)
```

Answer ???: 14

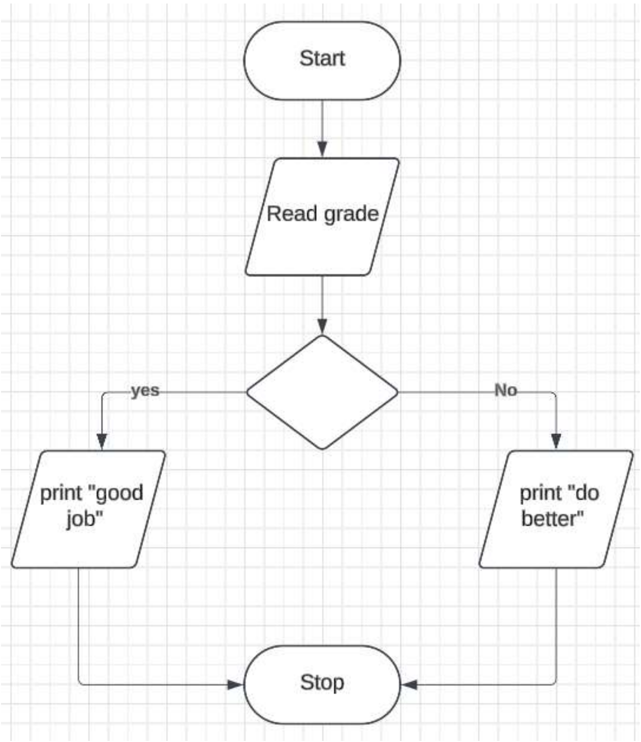
QB6)

```
s='ab cd ef'
s.casefold()
print(s.split())
```

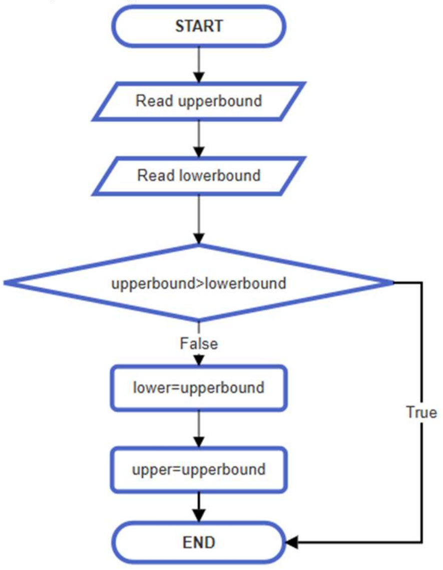
Answer ???: ['ab', 'cd', 'ef']

QB7) Complete the flowchart so that it describes the program segment given below.

```
grade = int(input('Enter grade:'))
if grade > 90:
    print('good job')
else:
    print('do better')
```



QB8) Complete the code for the following flowchart.



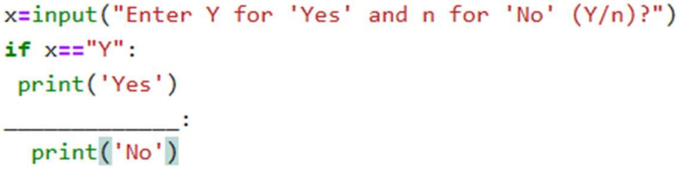
```
upperbound=int(input("Enter upper bound: "))
lowerbound=int(input("Enter lower bound: "))
if ______________________________:
    upper=upperbound
    lower=lowerbound
_________________:
    lower=upperbound
    upper=lowerbound
```

Answer 1???: upperbound>lowerbound

Answer 2 ????: else

PART C: Programming Questions (Each is 3pt)
MULTIPLE CHOICE (**Mark only one choice**)
(Each is 3pts)

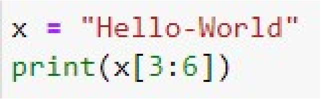
QC1) What should we fill in the blank?



```
x=input("Enter Y for 'Yes' and n for 'No' (Y/n)?")
if x=="Y":
 print('Yes')
____________:
  print('No')
```

[ ] elif x="n":
[ ] else x="n":
[ ] else x=="n":
[ ] elif x=="n": (*)

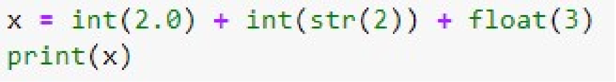
QC2) What is the output of the following piece of code?



```
x = "Hello-World"
print(x[3:6])
```

[ ] lo-W
[ ] lo- (*)
[ ] Wor
[ ] llo-

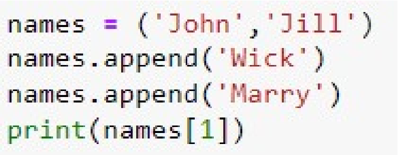
QC3) What is the output of the following piece of code?



```
x = int(2.0) + int(str(2)) + float(3)
print(x)
```

[ ] Error
[ ] 7
[ ] 7.0 (*)
[ ] float

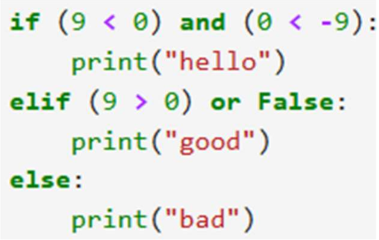
QC4) What is the output of the following piece of code?



```
names = ('John','Jill')
names.append('Wick')
names.append('Marry')
print(names[1])
```

[ ] Error (*)
[ ] ‘John’
[ ] ‘Wick’
[ ] ‘Jill’

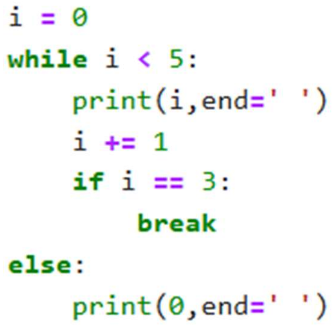
QC5) What is the output of the following piece of code?



```
if (9 < 0) and (0 < -9):
    print("hello")
elif (9 > 0) or False:
    print("good")
else:
    print("bad")
```

[ ] hello
[ ] good (*)
[ ] bad
[ ] none of the mentioned

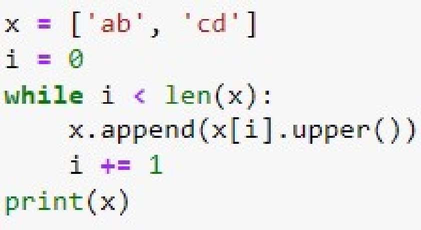
QC6) What is the output of the following code ?



```
i = 0
while i < 5:
    print(i,end=' ')
    i += 1
    if i == 3:
        break
else:
    print(0,end=' ')
```

[ ] 0 1 2 (*)
[ ] 0 1 2 0
[ ] 0 1 2 3
[ ] 0 1 2 3 0

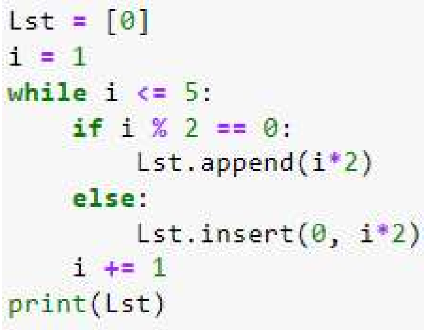
QC7) What are the final values of x?



```
x = ['ab', 'cd']
i = 0
while i < len(x):
    x.append(x[i].upper())
    i += 1
print(x)
```

[ ] ['ab', 'cd']
[ ] ['ab', 'cd', 'AB', 'CD']
[ ] ['AB', 'CD']
[ ] none of the mentioned (*)

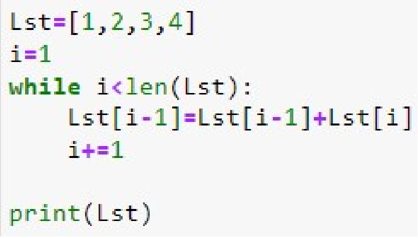
QC8) What is the output of the following code?



```
Lst = [0]
i = 1
while i <= 5:
    if i % 2 == 0:
        Lst.append(i*2)
    else:
        Lst.insert(0, i*2)
    i += 1
print(Lst)
```

[ ] [2, 4, 6, 8]
[ ] [4, 7, 5, 3]
[ ] [3, 5, 7, 4] (*)
[ ] [1, 2, 3, 4]

QC9) What is the output?



```
Lst=[1,2,3,4]
i=1
while i<len(Lst):
    Lst[i-1]=Lst[i-1]+Lst[i]
    i+=1

print(Lst)
```

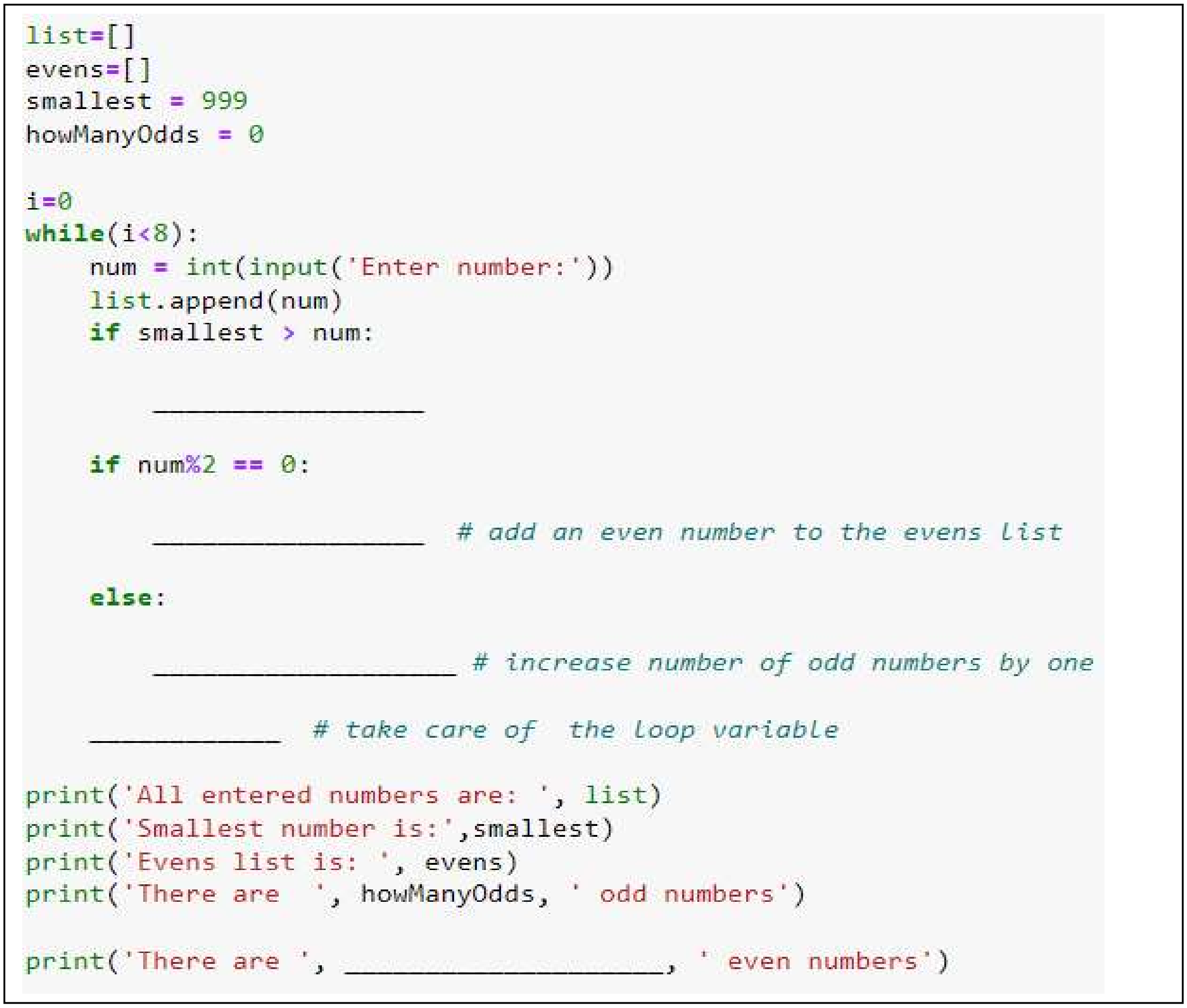
[ ] [2, 4, 6, 8]
[ ] [4, 7, 5, 3]
[ ] [3, 5, 7, 4] (*)
[ ] [1, 2, 3, 4]

PART D: CODING QUESTIONS

D1) (15 pts, 3 pts each slot)

Fill in the missing part of the Python program given below so that it performs the following actions:

- Read 8 numbers from the user. You can assume that numbers that are entered are less than 1000.
- Store all numbers in **list**
- Keep the smallest value so far in **smallest**
- Store all even numbers in **evens**
- Keep track of the number of odd numbers in **howManyOdds**
- Print all entered numbers, the smallest number, even numbers as a list, how many numbers are odd, how many numbers are even.



```
list=[]
evens=[]
smallest = 999
howManyOdds = 0

i=0
while(i<8):
    num = int(input('Enter number:'))
    list.append(num)
    if smallest > num:

        _________________

    if num%2 == 0:

        _________________  # add an even number to the evens list

    else:

        ___________________ # increase number of odd numbers by one

    ____________  # take care of  the loop variable

print('All entered numbers are: ', list)
print('Smallest number is:',smallest)
print('Evens list is: ', evens)
print('There are  ', howManyOdds, ' odd numbers')

print('There are ', ____________________, ' even numbers')
```

Sample output:

```
Enter number: 5
Enter number: 4
Enter number: 7
Enter number: 6
Enter number: 8
Enter number: 22
Enter number: 3
Enter number: 9
All entered numbers are:  [5, 4, 7, 6, 8, 22, 3, 9]
Smallest number is: 3
Evens list is:  [4, 6, 8, 22]
There are   4  odd numbers
There are  4 even numbers
```

QD2) (15pts)

A happy number is a number that eventually reaches 1 when replaced repeatedly by the sum of the square of each digit. If a number does not reach 1 and instead ends in a loop that never includes 1, it is called an unhappy number.

Steps to Determine if a number is Happy:

Start with a positive integer.

Replace the number by the sum of the squares of its digits.

Repeat the process until:

i.The number becomes 1, in which case it is a happy number.

ii.The number enters a cycle that does not include 1, indicating it is not a happy number.

Example:

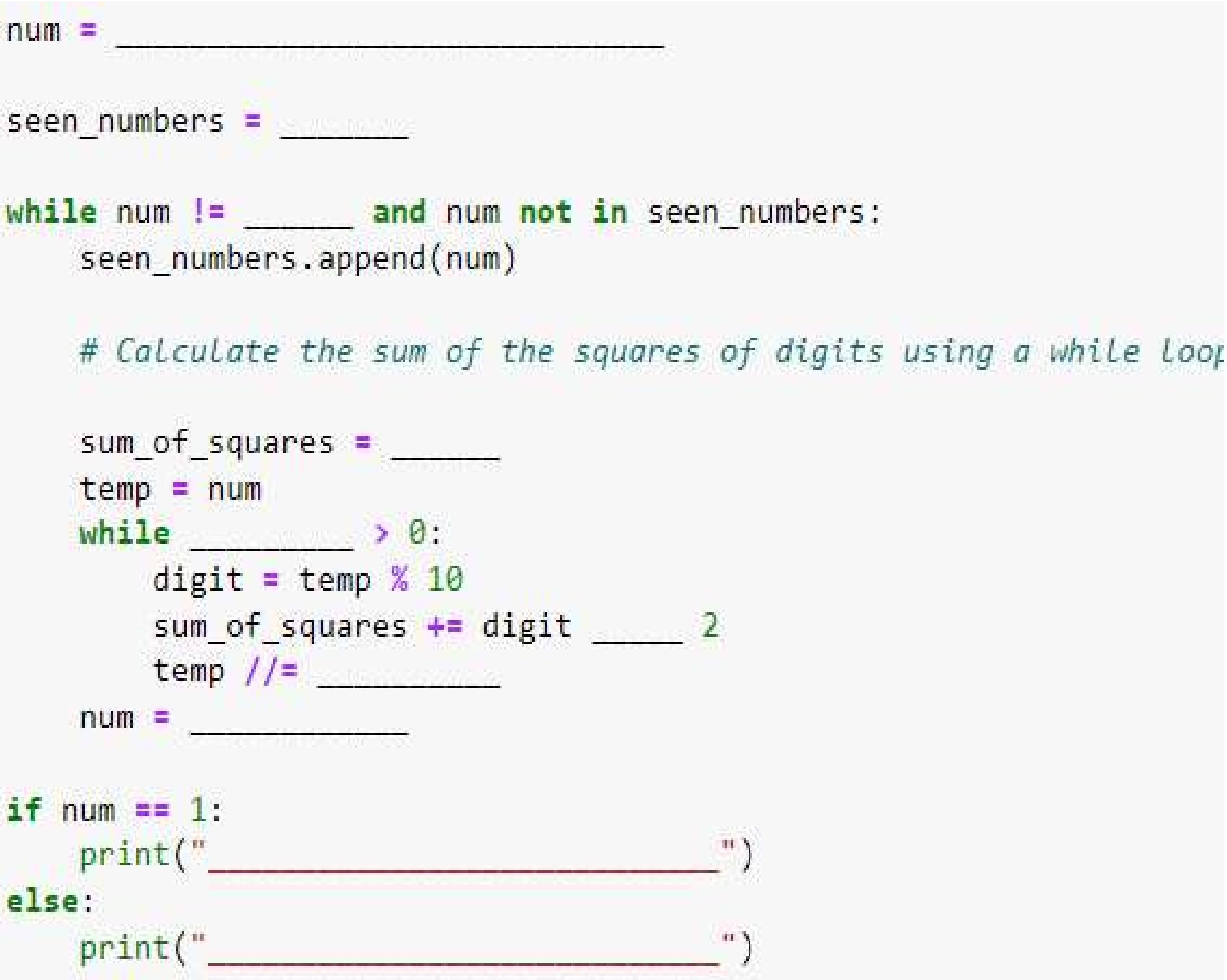
Input:13

$1^2+3^2=1+9=10$

$1^2+0^2=1+0=1$(The process reached1, so 13 is a happy number)

Fill in the missing part of the Python program given below so that it performs the following actions:

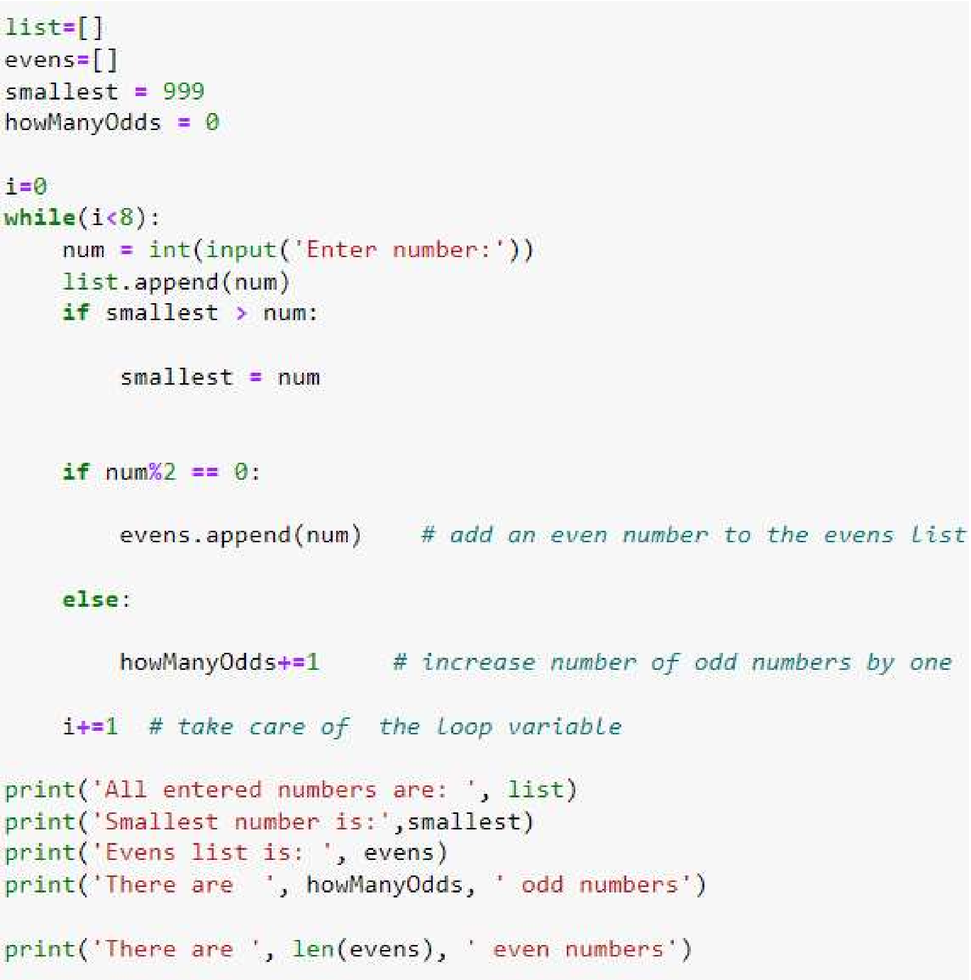
1. *Takes an integer input from the user.*
2. *Checks if the number is a happy number* or not.
3. Prints "Happy Number" if the number is happy, and "Not a Happy Number" if it is not.



```
num = ______________________________

seen_numbers = _______

while num != ______ and num not in seen_numbers:
    seen_numbers.append(num)

    # Calculate the sum of the squares of digits using a while loop

    sum_of_squares = ______
    temp = num
    while _________ > 0:
        digit = temp % 10
        sum_of_squares += digit _____ 2
        temp //= __________
    num = ____________

if num == 1:
    print("____________________________")
else:
    print("____________________________")
```

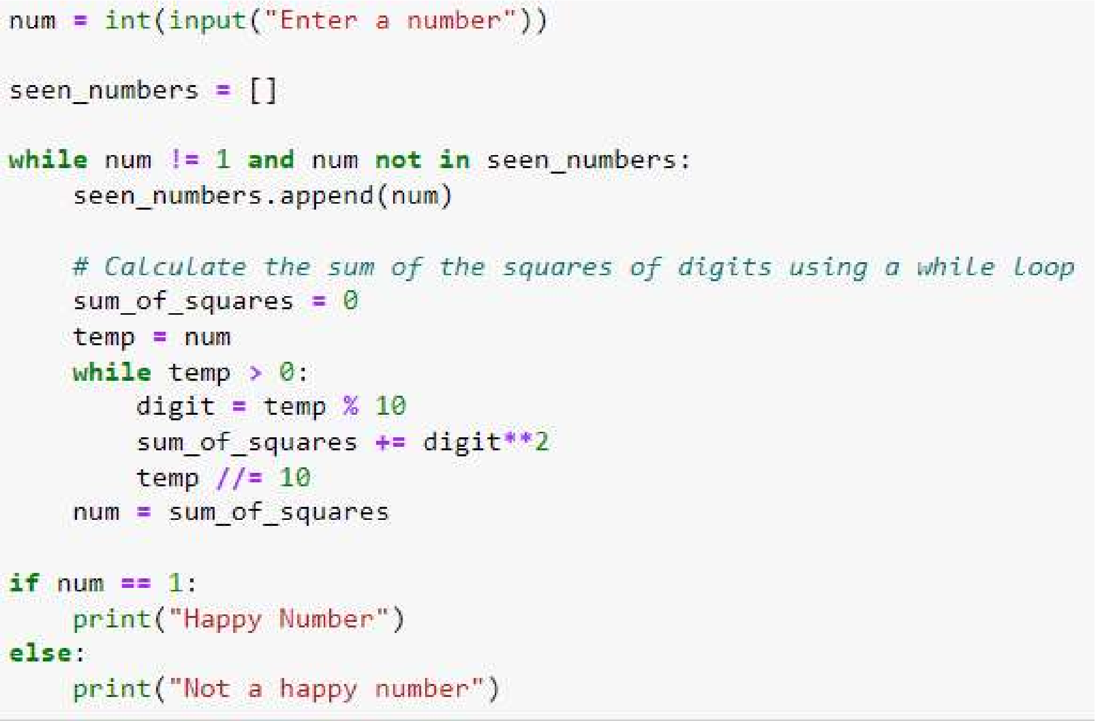
PART D SOLUTIONS

D1



```
list=[]
evens=[]
smallest = 999
howManyOdds = 0

i=0
while(i<8):
    num = int(input('Enter number:'))
    list.append(num)
    if smallest > num:

        smallest = num


    if num%2 == 0:

        evens.append(num)    # add an even number to the evens list

    else:

        howManyOdds+=1     # increase number of odd numbers by one

    i+=1  # take care of  the loop variable

print('All entered numbers are: ', list)
print('Smallest number is:',smallest)
print('Evens list is: ', evens)
print('There are  ', howManyOdds, ' odd numbers')

print('There are ', len(evens), ' even numbers')
```

D2



```
num = int(input("Enter a number"))

seen_numbers = []

while num != 1 and num not in seen_numbers:
    seen_numbers.append(num)

    # Calculate the sum of the squares of digits using a while loop
    sum_of_squares = 0
    temp = num
    while temp > 0:
        digit = temp % 10
        sum_of_squares += digit**2
        temp //= 10
    num = sum_of_squares

if num == 1:
    print("Happy Number")
else:
    print("Not a happy number")
```