

German University in Cairo  
Media Engineering and Technology Faculty  
Prof. Dr. Slim Abdennadher  
Dr. Aysha Elsafty

November 23, 2017

## CSEN102: Introduction to Computer Science Winter Semester 2017-2018 Midterm Exam

### Bar Code

**Instructions: Read carefully before proceeding.**

- 1) Duration of the exam: 2 hours (120 minutes).
- 2) No books or other aids are permitted for this test.
- 3) This exam booklet contains 11 pages, including this one. Three extra sheets of scratch paper are attached and have to be kept attached. **Note that if one or more pages are missing, you will lose their points. Thus, you must check that your exam booklet is complete.**
- 4) Write your solutions in the space provided. If you need more space, write on the back of the sheet containing the problem or on the three extra sheets and make an arrow indicating that. **Scratch sheets will not be graded unless an arrow on the problem page indicates that the solution extends to the scratch sheets.**
- 5) When you are told that time is up, stop working on the test.

**Good Luck!**

---

Don't write anything below ; -)

Exercise	1	2	3	4	5	6	$\Sigma$
Possible Marks	8	8	8	12	12	14	62
Final Marks							

**Exercise 1**

(8 Marks)

Beside each code fragment in the table below, write what is printed when the code fragment is executed. If the code would cause an error, write ERROR and give a brief explanation.

Code	Output or Cause of Error
<pre>x = 22 y = 33 x = y y = x print(x,y)</pre>	33 33
<pre>s = 'two' v = int(s) print(v)</pre>	Error. Cannot convert a string to an int
<pre>x = '1' + 2 print(x)</pre>	Error. Cannot concatenate an int to a string
<pre>while True:     print('Balabizo')     break</pre>	Balabizo
<pre>if True:     print('Balabizo')     break</pre>	Error. Invalid Syntax
<pre>x=float(5//2) print(x)</pre>	2.0
<pre>l = [2,3,4,5,1] print(l[l[len(l) - 1]])</pre>	3

**Exercise 2** Conditional Algorithms

(8 Marks)

For the first two years, a dog year is equal to 10.5 human years. After that, each dog year is equal to 4 human years.

Write a Python program to calculate a dog's age in dog years, given a dog's age in human years.

Expected output should look like the following:

- Input a dog's age in human years: 2  
The dog's age in dog years is 21
- Input a dog's age in human years: 15  
The dog's age in dog years is 73
- Input a dog's age in human years: -5  
Age must be positive number.

You **must** follow the printing format shown in the examples.

**Solution:**

```
h_age = int(input("Input a dog's age in human years: "))

if h_age < 0:
    print("Age must be positive number.")

elif h_age <= 2:
    d_age = h_age * 10.5
else:
    d_age = 21 + (h_age - 2)*4

print("The dog's age in dog's years is", d_age)
```

**Exercise 3** Conditional Algorithms

(8 Marks)

The following Python program displays the cooking time (in minutes) for a turkey of a given weight (in pounds) that may or may not be stuffed, according to the times in the following table:

Weight of turkey	Cooking time when not stuffed
under 14 pounds	195 minutes
14 to 20 pounds, inclusive	240 minutes
over 20 pounds	270 minutes

It adds 30 minutes to the cooking time if the turkey has been stuffed.

```
weight = eval(input())
stuffed = input()
if weight < 14:
    result = 195
if weight <= 20:
    result = 240
else:
    result = 270
if stuffed:
    result = result + 30
print(result)
```

The program should work for any input. You are asked to check whether the program is correct or not. If it is not correct, you should state why it is incorrect and re-write the program with the incorrect statement(s) fixed.

**Solution:**

The program is incorrect for the following reasons:

- `stuffed` should be evaluated as a boolean value upon taking it as input as it is used as such in the program
- For `weight < 14` the cooking time will be set to 195 and then re-set to 240

The correct program is:

```
weight = eval(input())
stuffed = eval(input())
if weight < 14:
    result = 195
elif weight <= 20:
    result = 240
else:
    result = 270
if stuffed:
    result = result + 30
print(result)
```

**Exercise 4** Iterative Algorithms

(12 Marks)

In mathematics, a geometric progression or geometric sequence is a sequence of numbers where each term after the first is found by multiplying the previous one by a fixed, non-zero number called the common ratio. For example, the sequence 2, 6, 18, 54, ... is a geometric progression with common ratio 3. Similarly, the sequence 10, 5, 2.5, 1.25, ... is a geometric progression with common ratio 1/2.

Write a Python program that takes a list of numbers representing a sequence and checks whether the sequence is a geometric progression or not. Your program should also detect and display the common ratio when applicable.

For example:

- If the input is the list [2, 6, 18, 54], then the output should be:

```
The sequence is a geometric progression with a common ratio of 3
```

- If the input is the list [10, 5, 2.5, 1.25], then the output should be:

```
The sequence is a geometric progression with a common ratio of 0.5
```

- If the input is the list [5, 8, 9, 11], then the output should be:

```
The sequence is not a geometric progression
Problem detected in index 2
```

- If the input is the list [22], then the output should be:

```
The sequence is a geometric progression
```

- If the input is an empty list, then the output should be:

```
There is no sequence
```

You **must** follow the printing format shown in the examples.

**Solution:**

```
x=eval(input())
if len(x)==0:
    print("There is no sequence")
elif len(x)==1:
    print("The sequence is a geometric progression")
else:
    ratio=x[1]/x[0]
    i=1
    while i<len(x)-1:
        if x[i+1]/x[i]!=ratio:
            break
        i+=1
    if i==len(x)-1:
        print("The sequence is a geometric progression with a common ratio of", ratio)
    else:
        print("The sequence is not a geometric progression")
        print("Problem detected in index", i+1)
```

**Solution of Exercise 4**

**Exercise 5** Iterative Algorithms

(12 Marks)

Write a Python program that takes a positive integer number as input and swaps the position of every  $(n + 1)^{th}$  digit with the  $n^{th}$  digit in the number.

If the number has an odd number of digits, then the first digit should not change its position.

For example,

- if the input number is 123456 then the expected output should be:

214365

- if the input number is 12345 then the expected output should be:

13254

**You are not allowed to use lists or Strings.**

**Solution:**

```
num=eval(input())
sum1=0
c=1
while num>0:
    _ if num>9:
        _ _ digit1= num%10
        _ _ num= int(num/10)
        _ _ digit2 = num%10
        _ _ sum1 = sum1 + (digit1*10 + digit2)*c
        _ _ c=c*100
    _ else:
        _ _ sum1 = sum1 + (num)*c
        _ num=int(num/10)

print(sum1)
```

**Exercise 6** Iterative Algorithms

(14 Marks)

A strobogrammatic number is a number whose numeral is rotationally symmetric, so that it appears the same when rotated 180 degrees. In other words, the numeral looks the same right-side up and upside down.

For example, given the number 69, if you look at it upside down (i.e. rotate it 180 degrees), it will still appear the same (69). This is due to the fact that the 9 upside down looks the same as 6, and the 6 upside down looks the same as a 9.

Write a Python program that, given a positive integer number, checks whether the number is a strobogrammatic number and displays a message accordingly.

For example,

- if the input number is 8008 then your program should display:

```
The number 8008 is a strobogrammatic number.
```

- if the input number is 19891 then your program should display:

```
The number 19891 is not a strobogrammatic number.
```

- if the input number is 19861 then your program should display:

```
The number 19861 is a strobogrammatic number.
```

- if the input number is 25 then your program should display:

```
The number 25 is not a strobogrammatic number.
```

You **must** follow the printing format shown in the examples.

**Solution:**

```
num=eval(input())
num1=num
num2=num
n=0
flag=True
while(num1>0):
    _ num1 = num1//10
    _ n+=1
    n-=1
while(num2>0):
    _ a=num2%10
    _ b=num2//(10**n)
    _ if(not((a==1 and b==1) or (a==8 and b==8) or (a==0 and b==0)
            or (a==6 and b==9) or (a==9 or b==6))):
        _ _ flag=False
    _ num2 = num2 - b*(10**n)
    _ num2 = num2//10
    _ n = n-2

if (flag):
    _ print("The num", num, " is a strobogrammatic number")
else:
    _ print("The num", num, " is not a strobogrammatic number")
```



---

**Scratch paper**

---

---

**Scratch paper**

---

---

**Scratch paper**

---