Week 12

COMP10001

Digital Ethics: Here are some of the points in the ACM (Association for Computing Machinery) Code of Ethics and Professional Conduct.

(a) Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing

(b) Avoid harm

(c) Be honest and trustworthy

(d) Be fair and take action not to discriminate

(e) Respect the work required to produce new ideas, inventions, creative works, and computing artefacts

(f) Respect privacy

(g) Honour confidentiality

Digital Ethics: Here are some of the points in the ACM (Association for Computing Machinery) Code of Ethics and Professional Conduct.

(a) Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing

(b) Avoid harm

(c) Be honest and trustworthy

(d) Be fair and take action not to discriminate

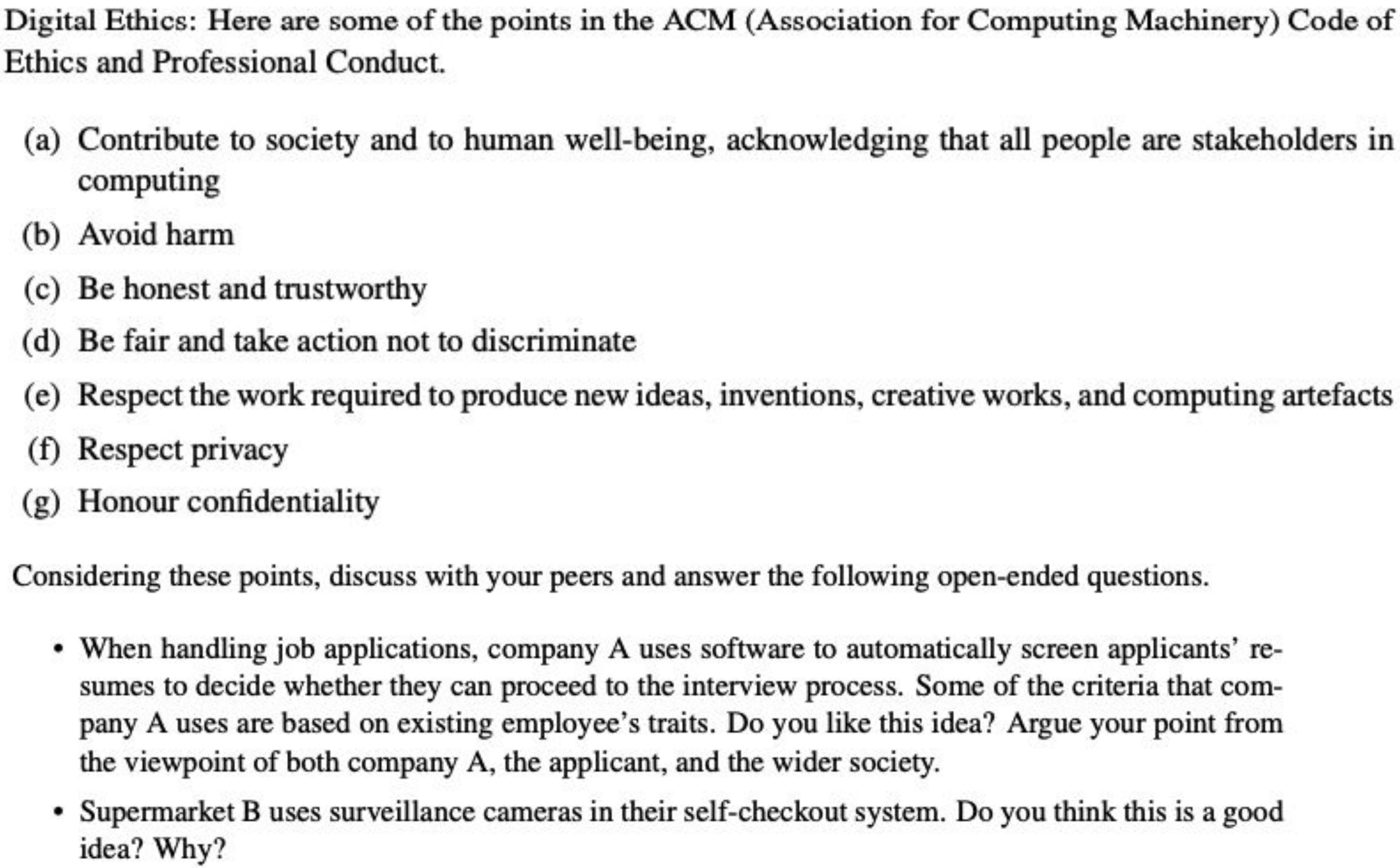
(e) Respect the work required to produce new ideas, inventions, creative works, and computing artefacts

(f) Respect privacy

(g) Honour confidentiality

Considering these points, discuss with your peers and answer the following open-ended questions.

- When handling job applications, company A uses software to automatically screen applicants' resumes to decide whether they can proceed to the interview process. Some of the criteria that company A uses are based on existing employee's traits. Do you like this idea? Argue your point from the viewpoint of both company A, the applicant, and the wider society.



Digital Ethics: Here are some of the points in the ACM (Association for Computing Machinery) Code of Ethics and Professional Conduct.

(a) Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing

(b) Avoid harm

(c) Be honest and trustworthy

(d) Be fair and take action not to discriminate

(e) Respect the work required to produce new ideas, inventions, creative works, and computing artefacts

(f) Respect privacy

(g) Honour confidentiality

Considering these points, discuss with your peers and answer the following open-ended questions.

- When handling job applications, company A uses software to automatically screen applicants' resumes to decide whether they can proceed to the interview process. Some of the criteria that company A uses are based on existing employee's traits. Do you like this idea? Argue your point from the viewpoint of both company A, the applicant, and the wider society.
- Supermarket B uses surveillance cameras in their self-checkout system. Do you think this is a good idea? Why?

Digital Ethics: Here are some of the points in the ACM (Association for Computing Machinery) Code of Ethics and Professional Conduct.

(a) Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing

(b) Avoid harm

(c) Be honest and trustworthy

(d) Be fair and take action not to discriminate

(e) Respect the work required to produce new ideas, inventions, creative works, and computing artefacts

(f) Respect privacy

(g) Honour confidentiality

Considering these points, discuss with your peers and answer the following open-ended questions.

- An online gambling website C is looking for an employee with a computing background. They want the employees to find patterns in customer online behaviours and provide personalised solutions for each customer. Doing so can generate more income for the website. One of your friends wants to apply for the job. What do you think of their decision?

Digital Ethics: Here are some of the points in the ACM (Association for Computing Machinery) Code of Ethics and Professional Conduct.

(a) Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing

(b) Avoid harm

(c) Be honest and trustworthy

(d) Be fair and take action not to discriminate

(e) Respect the work required to produce new ideas, inventions, creative works, and computing artefacts

(f) Respect privacy

(g) Honour confidentiality

Considering these points, discuss with your peers and answer the following open-ended questions.

- Student D is a part-time data analyst. Student D's employer gave them some confidential data to process to inform the business decisions. However, student D was right in the middle of the exam period. Due to the time pressure, they uploaded the data online and asked GenAI to do the task (such as ChatGPT), got the high-level analysis result, handled the result to their employer and claimed this was their own work. What are the problems of student D's approach?

What are the two criteria with which we can judge algorithms?

- Correctness
- Efficiency

What are the two criteria with which we can judge algorithms?

- Correctness
    - Does the algorithm give the correct output?
- Efficiency
    - How “good” is the algorithm?
        - Speed, storage, processing power, etc.
    - We haven’t looked at this in COMP10001! But it’s important to think about as you learn more about algorithms

What are the two criteria with which we can judge algorithms?

- Correctness
  - Does the algorithm give the correct output?
- Efficiency
  - How “good” is the algorithm?
    - Speed, storage, processing power, etc.
  - We haven’t looked at this in COMP10001! But it’s important to think about as you learn more about algorithms
- Which would you prefer? An algorithm guaranteed to calculate the correct answer that takes 150 years to finish, or an algorithm that takes seconds but may not always produce the correct result

What is the difference between exact and approximate approaches to algorithm design? Why might an approximate approach be necessary?

What is the difference between exact and approximate approaches to algorithm design? Why might an approximate approach be necessary?

- Exact approach: gives correct solution
- Approximate approach: gives almost the correct solution, through estimation, simulation, etc.

What is the difference between exact and approximate approaches to algorithm design? Why might an approximate approach be necessary?

- Exact approach: gives correct solution
- Approximate approach: gives almost the correct solution, through estimation, simulation, etc.
- If a problem is too complex to calculate with full completeness, an approximate approach might be more useful!

Some popular categories of algorithm

- Brute force (generate and test)
  - Exact. Finds every possible answer and tests it. Requires set of possible answers to be finite to guarantee completion. E.g. linear search
- Heuristic search
  - Approximate. E.g. finding the shortest path to a destination. Finding the definitive solution would require processing many possibilities - we can find an approximate search very quickly. (Ranks alternatives to prioritise possible paths to look at first) May find the exact solution!
- Simulation
  - Approximate. Finds a solution by generating lots of data to predict an overall trend. E.g. simulate play of a game to find out if it’s worth playing
- Divide and conquer
  - Exact. Divides problem into sub-problems which can be more easily solved. E.g. binary search

Search the following sorted lists for the number 8, using **(a)** Linear search (Brute-Force approach) and **(b)** Binary search (Divide and Conquer approach)

Think about the best, worst and average case scenarios of these algorithms. For example, can the best case scenario of a Brute-Force algorithm be faster than running the same task with a more clever algorithm?

(a)	1	2	4	5	8	9	10	12	15	19	21	23	25
(b)	8	9	11	15	16	17	22	24	27	28	29	32	33
(c)	2	4	5	6	7	9	11	12	13	15	19	22	25

Search the following sorted lists for the number 8, using **(a)** Linear search (Brute-Force approach) and **(b)** Binary search (Divide and Conquer approach)

Think about the best, worst and average case scenarios of these algorithms. For example, can the best case scenario of a Brute-Force algorithm be faster than running the same task with a more clever algorithm?

(a)	1	2	4	5	8	9	10	12	15	19	21	23	25
(b)	8	9	11	15	16	17	22	24	27	28	29	32	33
(c)	2	4	5	6	7	9	11	12	13	15	19	22	25

What is HTML? What’s it used for?

What is HTML? What’s it used for?

- HyperText Markup Language
- A markup language (as opposed to a programming language)

What is HTML? What’s it used for?

- HyperText Markup Language
- A markup language (as opposed to a programming language)
- Takes a document composed of text and other media and communicates how it should be rendered for display

What is HTML? What’s it used for?

- HyperText Markup Language
- A markup language (as opposed to a programming language)
- Takes a document composed of text and other media and communicates how it should be rendered for display
- E.g. an internet browser displays a webpage by downloading and presenting the website’s HTML

Revise the HTML tags we know!

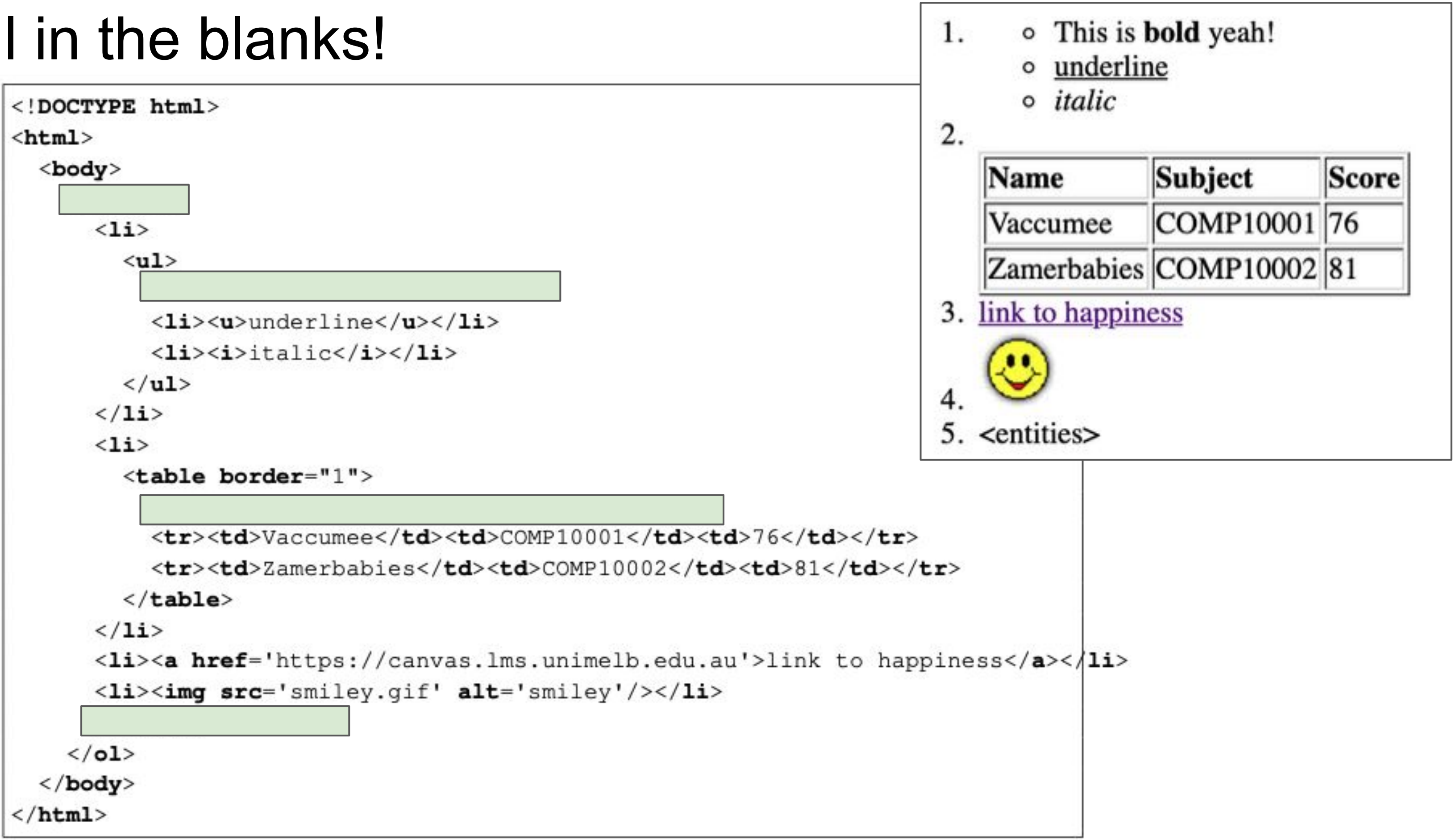
Revise the HTML tags we know!

- Use angle brackets: <open> </close>
- <b>, <i>, <u>
- <html>
- <head>, <body>
- <ul>, <ol>, <li>
- <table>, <tr>, <td>
- <a>, <img>, <audio>, <video>

Entities are special characters!

- An entity is a special character, written with &entity; syntax
- E.g. &lt; &gt; &nbsp; &amp;

Fill in the blanks!



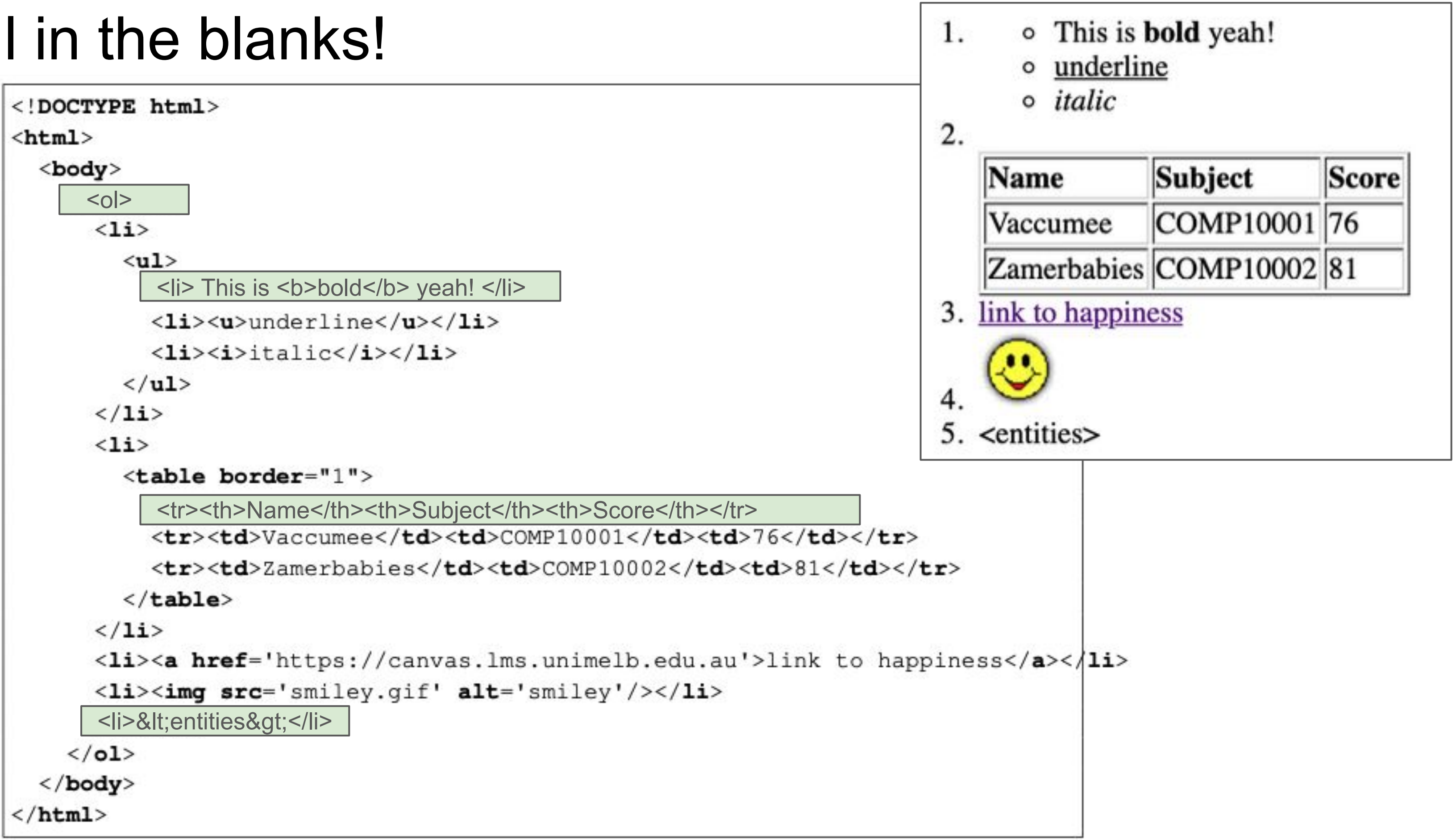
```
<!DOCTYPE html>
<html>
  <body>
    
      <li>
        <ul>
          
          <li><u>underline</u></li>
          <li><i>italic</i></li>
        </ul>
      </li>
      <li>
        <table border="1">
          
          <tr><td>Vaccumee</td><td>COMP10001</td><td>76</td></tr>
          <tr><td>Zamerbabies</td><td>COMP10002</td><td>81</td></tr>
        </table>
      </li>
      <li><a href='https://canvas.lms.unimelb.edu.au'>link to happiness</a></li>
      <li><img src='smiley.gif' alt='smiley'/></li>
     
    </ol>
  </body>
</html>
```

1.
   - This is **bold** yeah!
   - underline
   - *italic*
2.

Name	**Subject**	**Score**
Vaccumee	COMP10001	76
Zamerbabies	COMP10002	81

3. link to happiness
4.
5. <entities>

Fill in the blanks!



```
<!DOCTYPE html>
<html>
  <body>
    <ol>
      <li>
        <ul>
          <li> This is <b>bold</b> yeah! </li>
          <li><u>underline</u></li>
          <li><i>italic</i></li>
        </ul>
      </li>
      <li>
        <table border="1">
          <tr><th>Name</th><th>Subject</th><th>Score</th></tr>
          <tr><td>Vaccumee</td><td>COMP10001</td><td>76</td></tr>
          <tr><td>Zamerbabies</td><td>COMP10002</td><td>81</td></tr>
        </table>
      </li>
      <li><a href='https://canvas.lms.unimelb.edu.au'>link to happiness</a></li>
      <li><img src='smiley.gif' alt='smiley'/></li>
     <li>&lt;entities&gt;</li>
    </ol>
  </body>
</html>
```

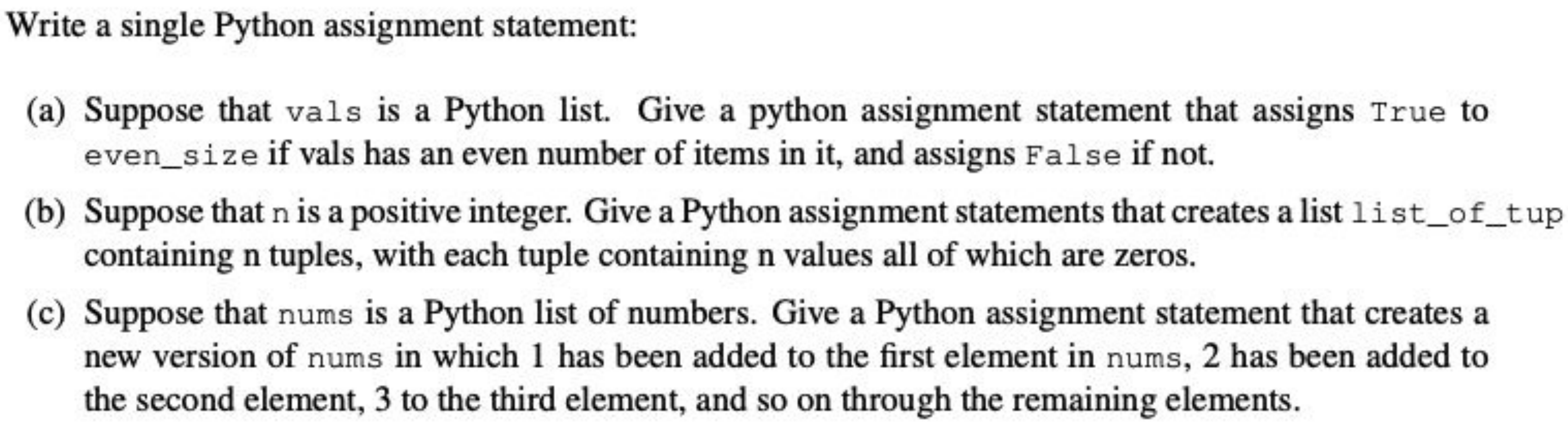
1.
   - This is **bold** yeah!
   - underline
   - *italic*
2.

Name	Subject	Score
Vaccumee	COMP10001	76
Zamerbabies	COMP10002	81

3. link to happiness
4.
5. <entities>

Let’s have a look at some HTML in the wild!

Past exam questions



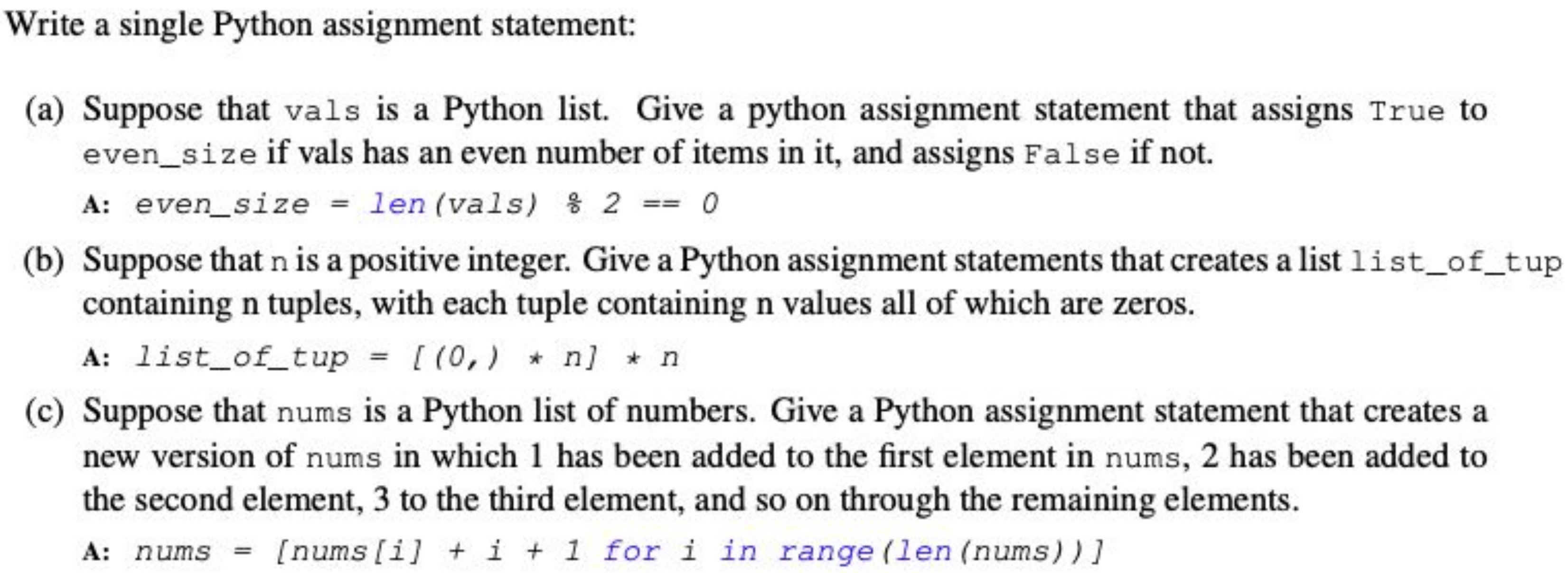
Write a single Python assignment statement:

(a) Suppose that `vals` is a Python list. Give a python assignment statement that assigns `True` to `even_size` if vals has an even number of items in it, and assigns `False` if not.

(b) Suppose that `n` is a positive integer. Give a Python assignment statements that creates a list `list_of_tup` containing n tuples, with each tuple containing n values all of which are zeros.

(c) Suppose that `nums` is a Python list of numbers. Give a Python assignment statement that creates a new version of `nums` in which 1 has been added to the first element in `nums`, 2 has been added to the second element, 3 to the third element, and so on through the remaining elements.

Past exam questions



Write a single Python assignment statement:

(a) Suppose that `vals` is a Python list. Give a python assignment statement that assigns `True` to `even_size` if vals has an even number of items in it, and assigns `False` if not.

A: `even_size = len(vals) % 2 == 0`

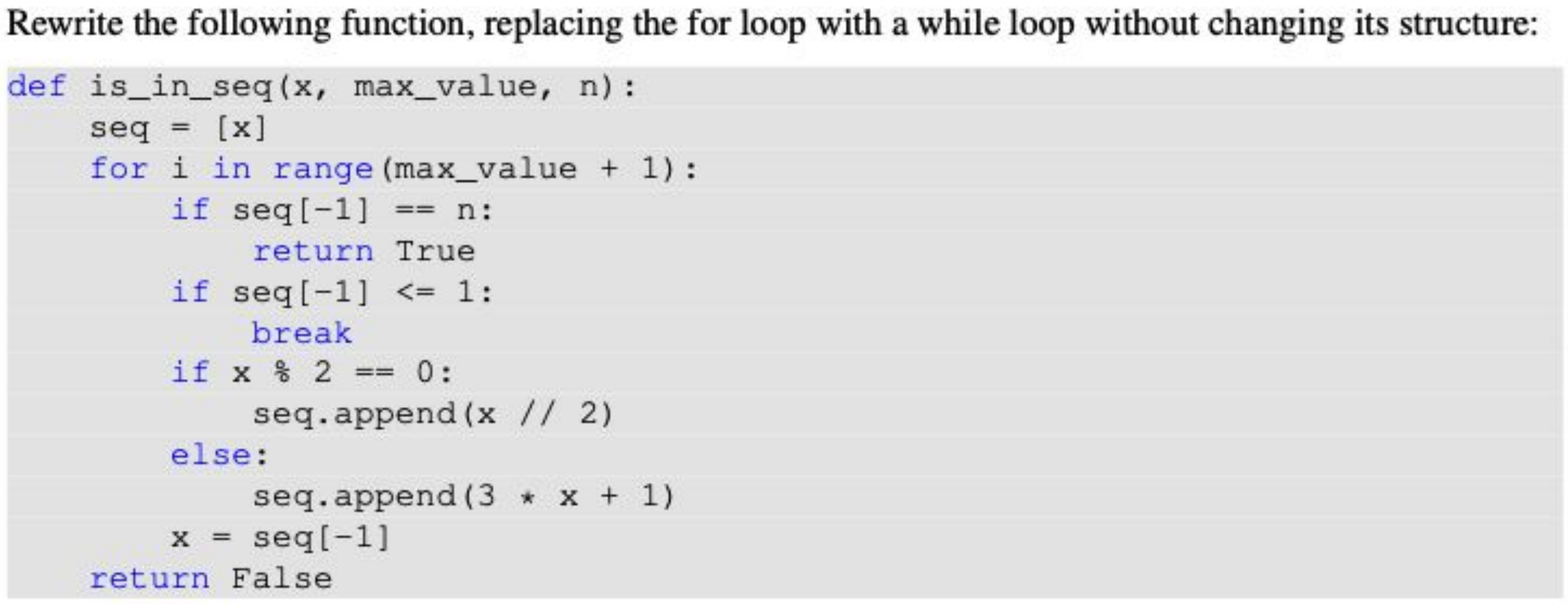
(b) Suppose that `n` is a positive integer. Give a Python assignment statements that creates a list `list_of_tup` containing n tuples, with each tuple containing n values all of which are zeros.

A: `list_of_tup = [(0,) * n] * n`

(c) Suppose that `nums` is a Python list of numbers. Give a Python assignment statement that creates a new version of `nums` in which 1 has been added to the first element in `nums`, 2 has been added to the second element, 3 to the third element, and so on through the remaining elements.

A: `nums = [nums[i] + i + 1 for i in range(len(nums))]`

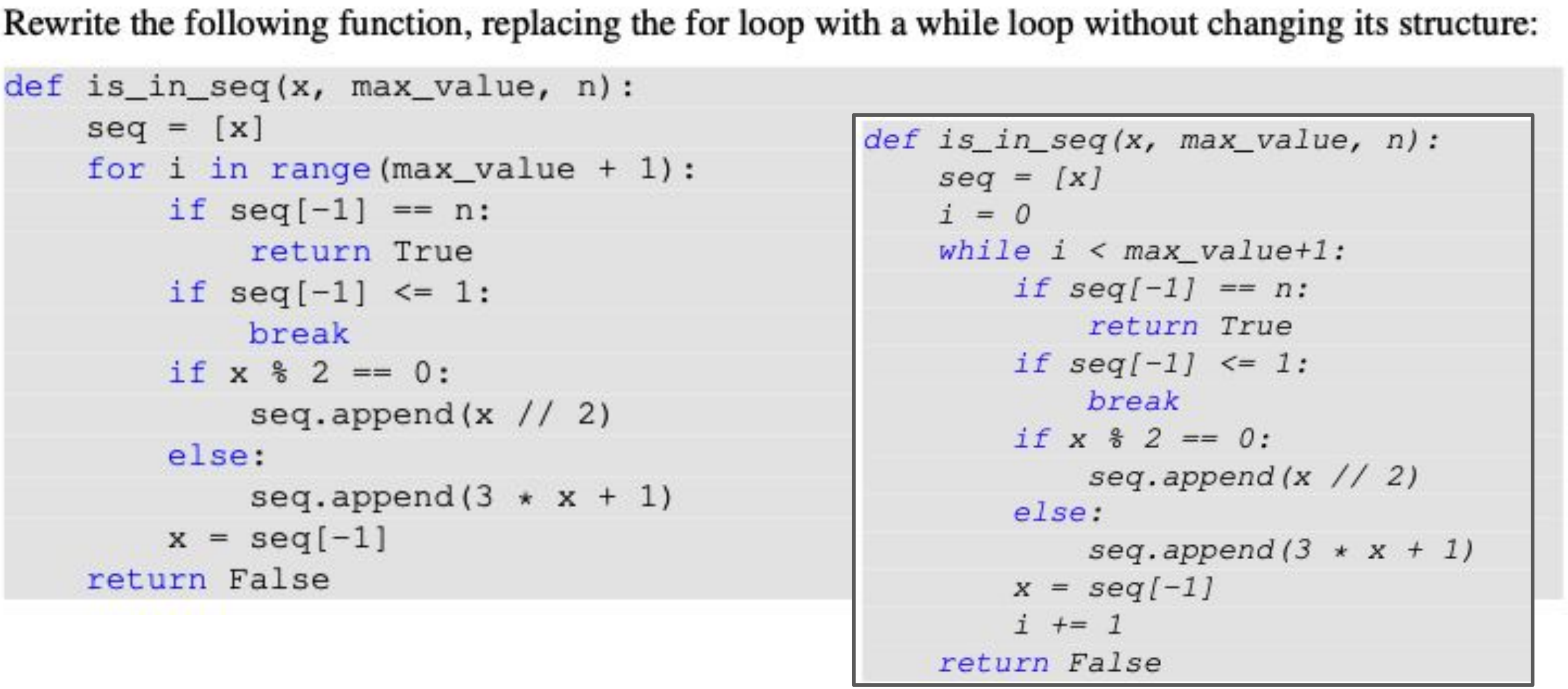
Past exam questions



Rewrite the following function, replacing the for loop with a while loop without changing its structure:

```
def is_in_seq(x, max_value, n):
    seq = [x]
    for i in range(max_value + 1):
        if seq[-1] == n:
            return True
        if seq[-1] <= 1:
            break
        if x % 2 == 0:
            seq.append(x // 2)
        else:
            seq.append(3 * x + 1)
        x = seq[-1]
    return False
```

Past exam questions

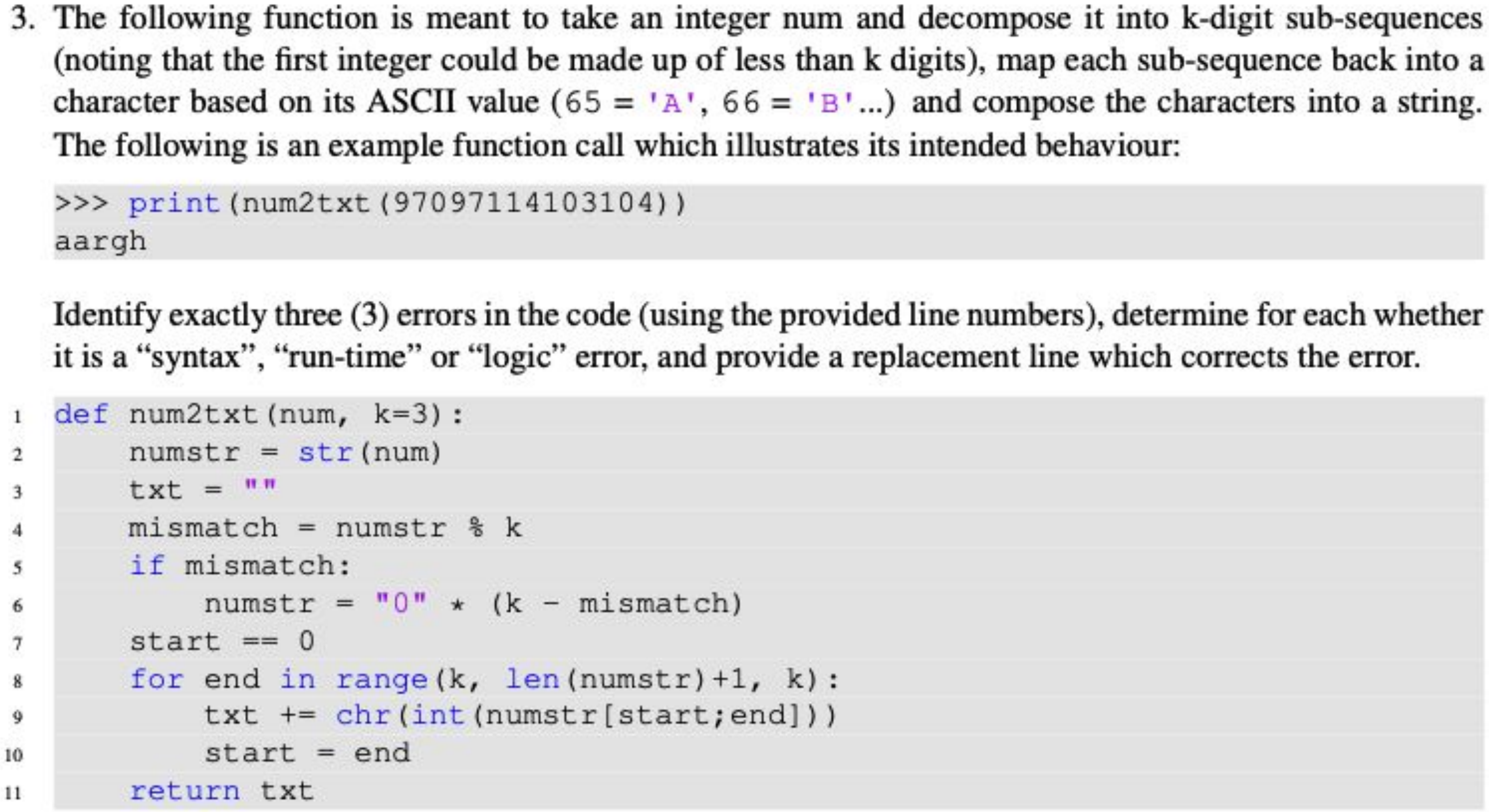


Rewrite the following function, replacing the for loop with a while loop without changing its structure:

```python
def is_in_seq(x, max_value, n):
    seq = [x]
    for i in range(max_value + 1):
        if seq[-1] == n:
            return True
        if seq[-1] <= 1:
            break
        if x % 2 == 0:
            seq.append(x // 2)
        else:
            seq.append(3 * x + 1)
        x = seq[-1]
    return False
```

```python
def is_in_seq(x, max_value, n):
    seq = [x]
    i = 0
    while i < max_value+1:
        if seq[-1] == n:
            return True
        if seq[-1] <= 1:
            break
        if x % 2 == 0:
            seq.append(x // 2)
        else:
            seq.append(3 * x + 1)
        x = seq[-1]
        i += 1
    return False
```

Past exam questions



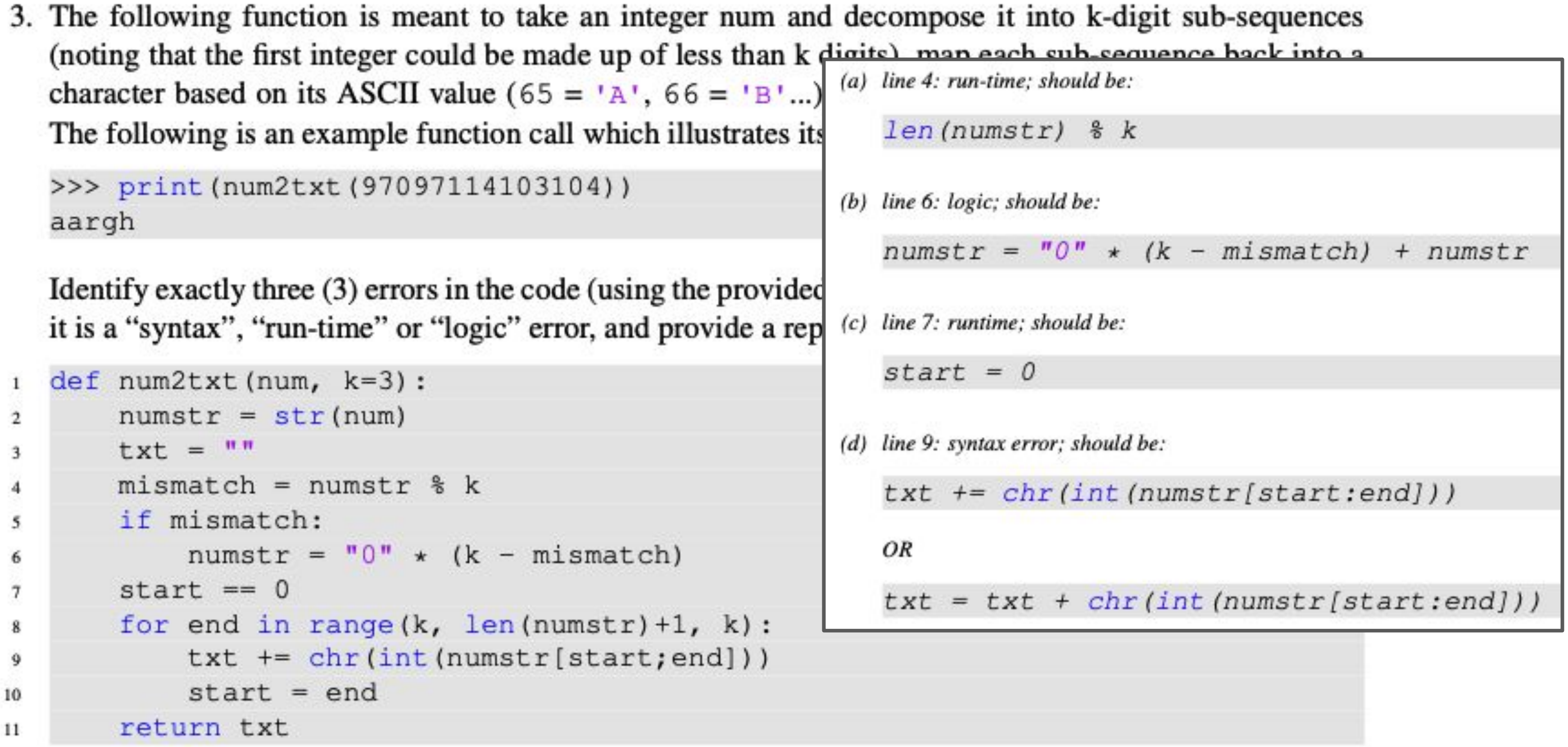
3. The following function is meant to take an integer num and decompose it into k-digit sub-sequences (noting that the first integer could be made up of less than k digits), map each sub-sequence back into a character based on its ASCII value (65 = 'A', 66 = 'B'...) and compose the characters into a string. The following is an example function call which illustrates its intended behaviour:

```
>>> print(num2txt(97097114103104))
aargh
```

Identify exactly three (3) errors in the code (using the provided line numbers), determine for each whether it is a "syntax", "run-time" or "logic" error, and provide a replacement line which corrects the error.

```
1  def num2txt(num, k=3):
2      numstr = str(num)
3      txt = ""
4      mismatch = numstr % k
5      if mismatch:
6          numstr = "0" * (k - mismatch)
7      start == 0
8      for end in range(k, len(numstr)+1, k):
9          txt += chr(int(numstr[start;end]))
10         start = end
11     return txt
```

Past exam questions



3. The following function is meant to take an integer num and decompose it into k-digit sub-sequences (noting that the first integer could be made up of less than k digits), map each sub-sequence back into a character based on its ASCII value (65 = 'A', 66 = 'B'...)
The following is an example function call which illustrates its

```
>>> print(num2txt(97097114103104))
aargh
```

Identify exactly three (3) errors in the code (using the provided
it is a "syntax", "run-time" or "logic" error, and provide a rep

```
1  def num2txt(num, k=3):
2      numstr = str(num)
3      txt = ""
4      mismatch = numstr % k
5      if mismatch:
6          numstr = "0" * (k - mismatch)
7      start == 0
8      for end in range(k, len(numstr)+1, k):
9          txt += chr(int(numstr[start;end]))
10         start = end
11     return txt
```

(a) *line 4: run-time; should be:*

```
len(numstr) % k
```

(b) *line 6: logic; should be:*

```
numstr = "0" * (k - mismatch) + numstr
```

(c) *line 7: runtime; should be:*

```
start = 0
```

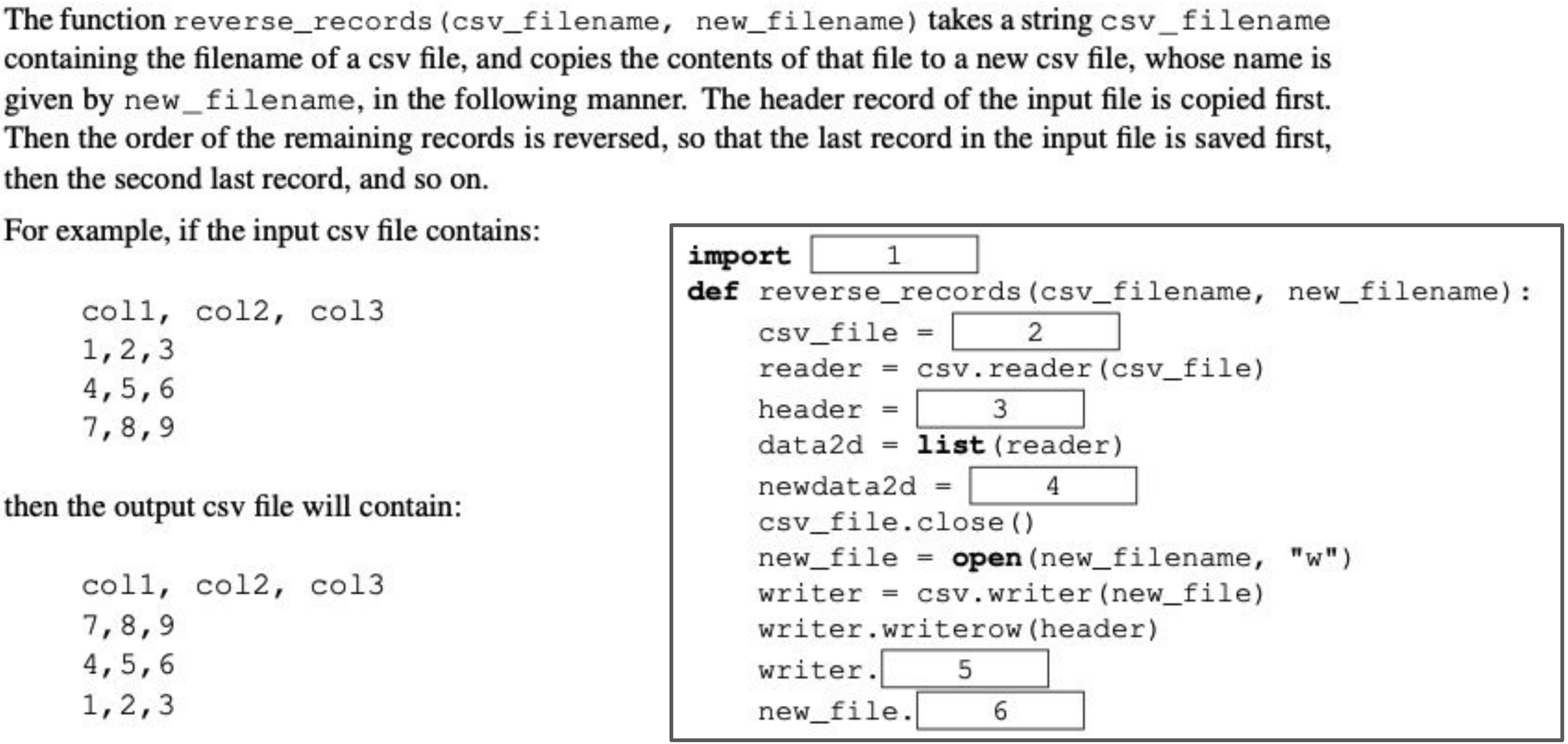
(d) *line 9: syntax error; should be:*

```
txt += chr(int(numstr[start:end]))
```

OR

```
txt = txt + chr(int(numstr[start:end]))
```

Past exam questions - fill in the blanks



The function `reverse_records(csv_filename, new_filename)` takes a string `csv_filename` containing the filename of a csv file, and copies the contents of that file to a new csv file, whose name is given by `new_filename`, in the following manner. The header record of the input file is copied first. Then the order of the remaining records is reversed, so that the last record in the input file is saved first, then the second last record, and so on.

For example, if the input csv file contains:

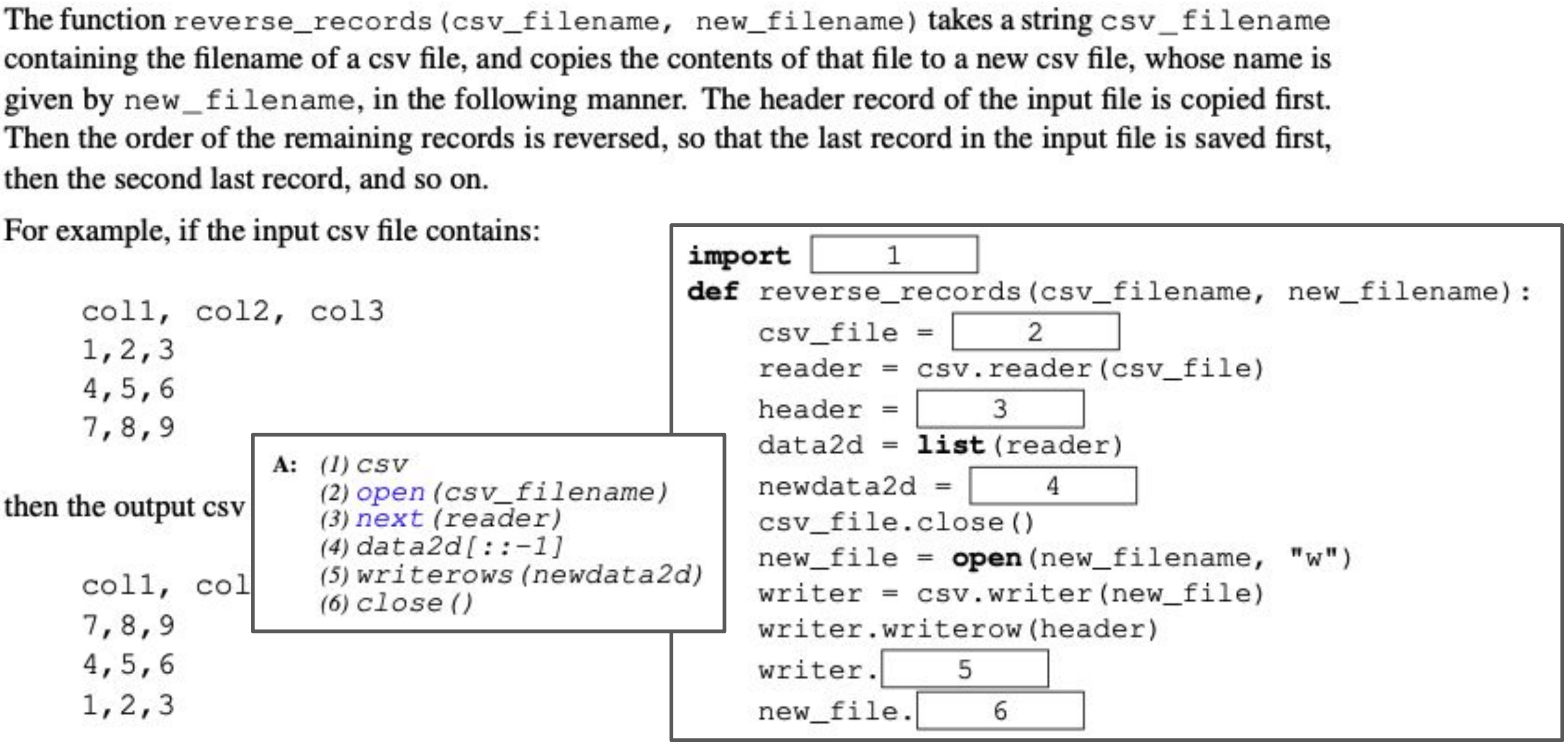
```
col1, col2, col3
1,2,3
4,5,6
7,8,9
```

then the output csv file will contain:

```
col1, col2, col3
7,8,9
4,5,6
1,2,3
```

```
import [    1    ]
def reverse_records(csv_filename, new_filename):
    csv_file = [    2    ]
    reader = csv.reader(csv_file)
    header = [    3    ]
    data2d = list(reader)
    newdata2d = [    4    ]
    csv_file.close()
    new_file = open(new_filename, "w")
    writer = csv.writer(new_file)
    writer.writerow(header)
    writer.[    5    ]
    new_file.[    6    ]
```

Past exam questions - fill in the blanks



The function `reverse_records(csv_filename, new_filename)` takes a string `csv_filename` containing the filename of a csv file, and copies the contents of that file to a new csv file, whose name is given by `new_filename`, in the following manner. The header record of the input file is copied first. Then the order of the remaining records is reversed, so that the last record in the input file is saved first, then the second last record, and so on.

For example, if the input csv file contains:

```
col1, col2, col3
1,2,3
4,5,6
7,8,9
```

then the output csv

```
col1, col
7,8,9
4,5,6
1,2,3
```

A: *(1)* `csv`
(2) `open(csv_filename)`
(3) `next(reader)`
(4) `data2d[::-1]`
(5) `writerows(newdata2d)`
(6) `close()`

```
import [    1    ]
def reverse_records(csv_filename, new_filename):
    csv_file = [    2    ]
    reader = csv.reader(csv_file)
    header = [    3    ]
    data2d = list(reader)
    newdata2d = [    4    ]
    csv_file.close()
    new_file = open(new_filename, "w")
    writer = csv.writer(new_file)
    writer.writerow(header)
    writer.[    5    ]
    new_file.[    6    ]
```

Short answer questions:

(a) Suppose you are generating all possible permutations to find the best score in some problem. Is this an approximate or an exact approach? Briefly justify your answer.

(b) Using an example, briefly explain what heuristics are and why we need them.

5. Short answer questions:

   (a) Suppose you are generating all possible permutations to find the best score in some problem. Is this an approximate or an exact approach? Briefly justify your answer.

   A: *Exact approach. This is an example of a brute-force strategy, in which all possible answers are generated and tested to find the result.*

   (b) Using an example, briefly explain what heuristics are and why we need them.

   A: *One example is to use Euclidean distance as the heuristic to find the nearest cinema from the university. The Euclidean distance is easy to compute, hence it is often regarded "close enough" and efficient, especially when finding the definitive solution is significantly slower and more complicated.*