There are five problems listed below. To get full credit for this assignment you need to complete all of them.

If you are stuck or confused by any of the problems, please ask your tutor, lecturer or post to Piazza. You are allowed to discuss the problems with your peers and refer to online materials, but you are not allowed to share solutions or copy materials from any source. You may find the academic integrity rules at `https://academicintegrity.cs.auckland.ac.nz/`.

To get full marks you need to **show all working** unless a question explicitly states not to.

You should submit via Canvas a single PDF file containing the answers to the written questions. A scanned handwritten submission is acceptable if it is neatly written (if it's hard to read, it will not be marked). If typing the assignment, do the best you can with mathematical symbols. For exponents, write something like `2^n` if using plain text. Use LaTeX if you really want it to look good.

Answers to programming questions must be submitted via the automated marker system at `https://www.automarker.cs.auckland.ac.nz/student.php`.

Please try to submit your assignments no later than 5 min before the due time. Late assignments will not be accepted. If you need an extension due to illness or other personal circumstance, please email marc.vinyals@auckland.ac.nz as early as possible **before** the due date.

Best of luck, and enjoy the problems!

Problem 1 (10 marks)**.** You are given 3 algorithms A_1,A_2 and A_3 solving the same problem. Their running times are

$$T_1(n) = 10^8 \cdot n^4 + n^8, \qquad T_2(n) = 10^n + 10^8, \qquad T_3(n) = 10^n + \log_{10}(n)$$

a) Which algorithm is the fastest for very large inputs? Which algorithm is the slowest for very large inputs? (Justify your answer.)

b) For which input sizes is A_2 faster than A_1? (Justify your answer.)

c) For which input sizes is A_3 faster than A_2? (Justify your answer.)

Assume that input sizes are always positive integers.

Problem 2 (10 marks)**.** Write down a recurrence relation that describes the number of elementary operations in the worst case for the following algorithm (justify your answer). Variable assignments, comparisons, arithmetic operations, and function calls are considered to be elementary, but you do not need to count memory accesses or function returns. You do not need to solve the recurrence.

n is a positive integer. Lists a and b are a global variables, which have at least n elements.

```
Function ldequere($n$):
    if $n < 9$ then
        return 0
    for $i \leftarrow 1$ to $n$ do
        $b[i] \leftarrow a[i]$
    for $i \leftarrow 9$ to $n$ do
        if $b[i-8] = 0$ then
            return $i$
        $b[i] \leftarrow 0$
    return ldequere($n-1$)
```

Problem 3 (10 marks). In this problem you have to prove facts about the asymptotic behaviour of some functions. It will be more convenient to either use the definitions directly or the rules for asymptotic notation, depending on the question.

a) Prove that $\log_2(n^8)$ is $O(\log_2(n^4))$.

b) Prove that n^8 is not $O(n^4)$.

c) Prove that $\cos(n)+2$ is $\Omega(10)$.
Assume that n is an angle in degrees, so for example $\cos(90)=0$.

d) Prove that $\cos(n)+1$ is not $\Omega(10)$.
Hint: proof by contradiction.

e) Let p be a function in $\Theta(2^n)$, let q be a function in $O(3^n)$, let r be a function in $\Omega(4^n)$, and let $s=pq+pr$. In other words, $s(n)=p(n)\cdot q(n)+p(n)\cdot r(n)$. What can you say about function s? (write everything that you can deduce, and justify your answer.)

Problem 4 (10 marks). Calculate the number of elementary operations in the worst case for the following algorithm (justify your answer). You only need to count operations stated explicitly as "C elementary operations", and you may ignore arithmetic operations, comparisons, etc.

You may assume that n is an integer of the form 2^{2k}. C is a constant.

```
Function spopsacr($n$):
    for $i \leftarrow 1$ to $2n$ do
        for $j \leftarrow i$ to $\log n$ do
            $C$ elementary operations
    for $i \leftarrow 1$ to $n$ do
        if $i$ is a perfect square then
            $C$ elementary operations
        else
            for $j \leftarrow 1$ to $n$ do
                $C$ elementary operations
    $i \leftarrow 2$
    while $i < n$ do
        $i \leftarrow i \cdot i$
        $C$ elementary operations
```

Problem 5 (10 marks)**.** Write a programme that takes input as described below and prints output as described below. The programme must work with the automated marker. This question is purely for you to obtain familiarity with the automated marker system which will be used in later assignments.

Input: The input consists of a number of lines. For example:

```
blah
45 67
ddgfh fjhg gjkhgk
```

Output: The output must consist of the same lines as the input, each line reversed. For example, for the input given above the output should be:

```
halb
76 54
kghkjg ghjf hfgdd
```

Access the automarker via `https://www.automarker.cs.auckland.ac.nz/student.php`.

You must submit your solution as a Python programme.

There is a limit of 10 submission attempts for this assignment in order to get full marks. The last submission submitted before the assignment deadline will be the one marked. Beyond 10 submissions, a penalty will apply, but every student who eventually submits a correct answer on time will get 75% for this question. In future assignments, restrictions and penalties may be stronger.

You can assume that input will come from standard input (stdin) in a stream that represents one string per line. Output should be sent to standard output (stdout). In the automarker, the input will come from a file that is piped to standard in and the output redirected to a file, but your programme shouldn't know that.

Your code should be contained in a single file. You may assume that the automarker has access to all standard libraries.

If your submission was put in the queue before the assignment due time then it will be accepted. Submissions after the assignment due time will not be considered.

Start early! Lots of students will be submitting their work closer to the deadline so it might take 30 min before your programme is executed and you get to see the results.

Your output should exactly match the one in the system for the submission to be correct. So be careful with the printing. No extra symbols! It may look the same on the first glance but may have a different end of line character or extra space.

Please test your programme locally before submitting it. You may use a command sequence like the following.

```
$ python3 task1.py < sample.in > my.out
$ diff my.out sample.out
```