Assignment 1 – Discrete Review

Answer the questions in the boxes provided on the question sheets. If you run out of room for an answer, add a page to the end of the document.

Related Readings: `http://pages.cs.wisc.edu/~hasti/cs240/readings/`

Name: KJ Choi

Wisc id: 908 028 9540

Logic

1. Using a truth table, show the equivalence of the following statements.

   (a) $P \vee (\neg P \wedge Q) \equiv P \vee Q$

Solution:

P	Q	$\neg P$	$\neg P \wedge Q$	$P \vee (\neg P \wedge Q)$	$P \vee Q$
T	T	F	F	T	T
T	F	F	F	T	T
F	T	T	T	T	T
F	F	T	F	F	F

   (b) $\neg P \vee \neg Q \equiv \neg(P \wedge Q)$

Solution:

P	Q	$\neg P$	$\neg Q$	$P \wedge Q$	$\neg P \vee \neg Q$	$\neg(P \wedge Q)$
T	T	F	F	T	F	F
T	F	F	T	F	T	T
F	T	T	F	F	T	T
F	F	T	T	F	T	T

(c) $\neg P \lor P \equiv \text{true}$

Solution:

P	$\neg P$	$\neg P \lor P$	True
T	F	T	T
F	T	T	T

(d) $P \lor (Q \land R) \equiv (P \lor Q) \land (P \lor R)$

Solution:

P	Q	R	$Q \land R$	$P \lor Q$	$P \lor R$	$P \lor (Q \land R)$	$(P \lor Q) \land (P \lor R)$
T	T	T	T	T	T	T	T
T	T	F	F	T	T	T	T
T	F	T	F	T	T	T	T
T	F	F	F	T	T	T	T
F	T	T	T	T	T	T	T
F	T	F	F	T	F	F	F
F	F	T	F	F	T	F	F
F	F	F	F	F	F	F	F

Sets

2. Based on the definitions of the sets A and B, calculate the following: $|A|$, $|B|$, $A \cup B$, $A \cap B$, $A \setminus B$, $B \setminus A$.

   (a) $A = \{1, 2, 6, 10\}$ and $B = \{2, 4, 9, 10\}$

   Solution:
   $|A| = 4$
   $|B| = 4$
   $A \cup B = \{1, 2, 4, 6, 9, 10\}$
   $A \cap B = \{2, 10\}$
   $A \setminus B = \{1, 6\}$
   $B \setminus A = \{4, 9\}$

   (b) $A = \{x \mid x \in \mathbb{N}\}$ and $B = \{x \in \mathbb{N} \mid x \text{ is even}\}$

   Solution:
   $|A| = \infty$
   $|B| = \infty$
   $A \cup B = \{x \mid x \in \mathbb{N}\}$
   $A \cap B = \{x \in \mathbb{N} \mid x \text{ is even}\}$
   $A \setminus B = \{x \in \mathbb{N} \mid x \text{ is odd}\}$
   $B \setminus A = \varnothing$

Relations and Functions

3. For each of the following relations, indicate if it is reflexive, antireflexive, symmetric, antisymmetric, or transitive.

   (a) $\{(x, y) : x \leq y\}$

   Solution:
   reflexive, transitive

   (b) $\{(x, y) : x > y\}$

   Solution:
   antireflexive, transitive

(c) $\{(x, y) : x < y\}$

Solution: antireflexive, transitive

(d) $\{(x, y) : x = y\}$

Solution: reflexive, symmetric, transitive

4. For each of the following functions (assume that they are all $f : \mathbb{Z} \to \mathbb{Z}$), indicate if it is surjective (onto), injective (one-to-one), or bijective.

(a) $f(x) = x$

Solution: bijective

(b) $f(x) = 2x - 3$

Solution: bijective

(c) $f(x) = x^2$

Solution: onto

5. Show that $h(x) = g(f(x))$ is a bijection if $g(x)$ and $f(x)$ are bijections.

Solution:

In order to prove $h(x) = g(f(x))$ is a bijection function given that $g(x)$ and $f(x)$ are bijections, we need to prove that the function is injective (i) and surjective (ii).

To prove the the function is injective (i), we must prove that $(\forall a_1, a_2)h(a_1) = h(a_2) \to a_1 = a_2$.
We can rewrite the statement $h(a_1) = h(a_2)$ as $g(f(a_1)) = g(f(a_2))$.
Since g and f are bijective, $g(f(a_1)) = g(f(a_2)) \Rightarrow f(a_1) = f(a_2) \to a_1 = a_2$.
Thus, $(\forall a_1, a_2)h(a_1) = h(a_2) \to a_1 = a_2$ is true, i.e. $h(x)$ is injective.

To prove the the function is surjective (ii), we must prove that $(\forall c \in C)(\exists a \in A)(h(a) = g(f(a)) = c)$, given that C and A are domain and range of the function $h(x)$.
Let $f : A \to B$ and $g : B \to C$. Since f and g are bijective, $(\forall b \in B)(\exists! a \in A)(f(a)) = b)$ and $(\forall c \in C)(\exists! b \in B)(g(b)) = c)$. This implies $(\forall c \in C)(\exists! a \in A)(h(a) = g(f(a)) = c)$.
Thus, $(\forall c \in C)(\exists a \in A)(h(a) = c)$ holds true, i.e. $h(x)$ is surjective.

Since $h(x)$ is both injective (i) and surjective (ii), $h(x)$ is bijective.

Induction

6. Prove the following by induction.

(a) $\sum_{i=1}^{n} i = n(n+1)/2$

Solution:
We use induction to show that $P(n)$ holds for all natural numbers $n \geq 1$ where
$P(n) : \sum_{i=1}^{n} i = n(n+1)/2$

Base case: $P(1) : \sum_{i=1}^{1} i = 1$ and $1(1+1)/2 = 1$. So, $P(1)$ holds.

Inductive case: IH: $P(k)$ holds, i.e. $\sum_{i=1}^{k} i = k(k+1)/2$
Now consider $P(k+1) : \sum_{i=1}^{k+1} i = \sum_{i=1}^{k} i + (k+1) = k(k+1)/2 + (k+1)$
by induction hypothesis.
$k(k+1)/2 + (k+1) = (k+1)(k+2)/2 = (k+1)((k+1)+1)/2$. Thus, $P(k+1)$ holds.

Therefore, by induction, $P(n)$ holds for all natural number $n \geq 1$.

(b) $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$

Solution:
Let $P(n) : \sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$ for all natural numbers $n \geq 1$.

Base case: $P(1) : \sum_{i=1}^{1} i^2 = 1$ and $1(1+1)(2*1+1)/6 = 6/6 = 1$. So, $P(1)$ holds.

Inductive case: IH: $P(k)$ holds, i.e. $\sum_{i=1}^{k} i^2 = k(k+1)(2k+1)/6$
Now consider $P(k+1) : \sum_{i=1}^{k+1} i^2 = \sum_{i=1}^{k} i^2 + (k+1)^2 = k(k+1)(2k+1)/6 + (k+1)^2$
by induction hypothesis.
$k(k+1)(2k+1)/6 + (k+1)^2 = (k+1)(2k^2+7k+6)/6 = (k+1)(k+2)(2k+3)/6$
$= (k+1)((k+1)+1)(2(k+1)+1)/6$. Thus, $P(k+1)$ holds.

Therefore, by induction, $P(n)$ holds for all natural number $n \geq 1$.

(c) $\sum_{i=1}^{n} i^3 = n^2(n+1)^2/4$

Solution:
Let $P(n) : \sum_{i=1}^{n} i^3 = n^2(n+1)^2/4$ for all natural numbers $n \geq 1$.

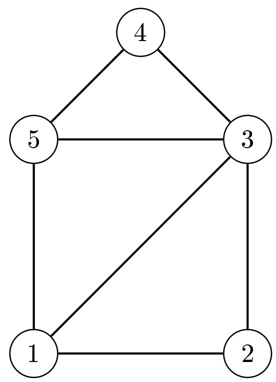
Base case: $P(1) : \sum_{i=1}^{1} i^3 = 1$ and $1^2(1+1)^2/4 = 1$. So, $P(1)$ holds.

Inductive case: IH: $P(k)$ holds, i.e. $\sum_{i=1}^{k} i^3 = n^2(n+1)^2/4$
Now consider $P(k+1) : \sum_{i=1}^{k+1} i^3 = \sum_{i=1}^{k} i^3 + (k+1)^3 = k^2(k+1)^2/4 + (k+1)^3$
by induction hypothesis.
$k^2(k+1)^2/4 + (k+1)^3 = (k+1)^2(k^2+4k+4)/4 = (k+1)^2(k+2)^2/4$
$= (k+1)^2((k+1)+1)^2/4$
Thus, $P(k+1)$ holds.

Therefore, by induction, $P(n)$ holds for all natural number $n \geq 1$.

Graphs and Trees

7. Give the adjacency matrix, adjacency list, edge list, and incidence matrix for the following graph.



Solution:

adjacency matrix: $\begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$

adjacency list:

1	[2, 3, 5]
2	[1, 3]
3	[1, 2, 4, 5]
4	[3, 5]
5	[1, 3, 4]

edge list: [(1,2), (1,3), (1,5), (2,3), (3,4), (3,5), (4,5)]

incidence matrix: $\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$

8. How many edges are there is a complete graph of size n? Prove by induction.

Solution:

Let $P(n) : |E_n| = n(n-1)/2$, given $n \geq 1$ and E is an edge list of a complete graph of size n.

Base case: $P(1) : 1(1-1)/2 = 0$ and a complemte graph with single node has no edges. So $P(1)$ holds.

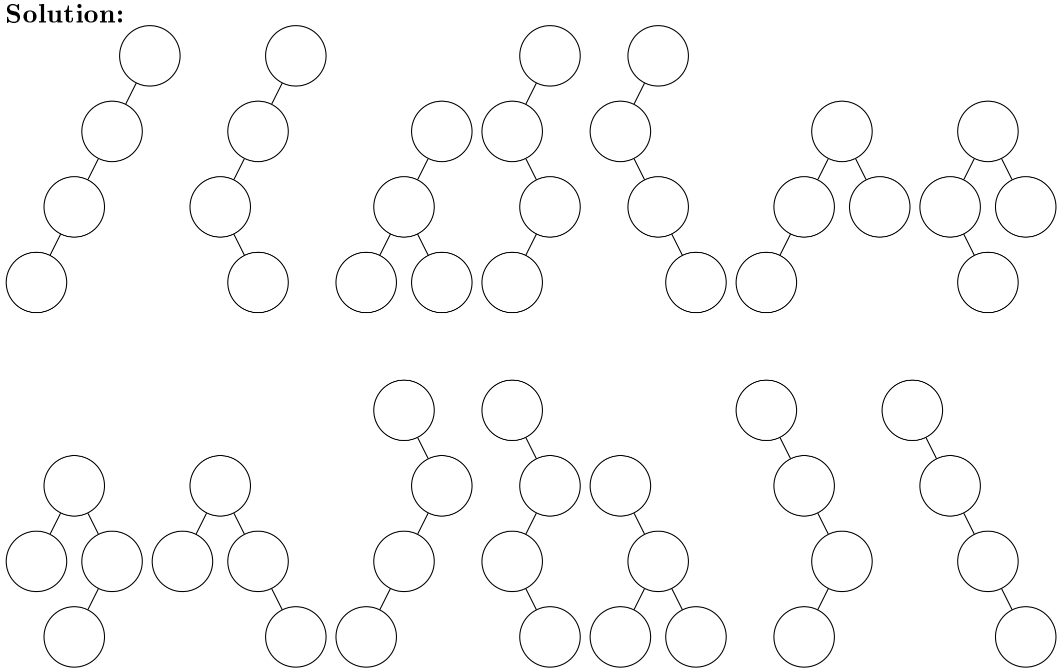
Inductive case: IH: $P(k)$ holds, i.e. $|E_k| = k(k-1)/2$

Now consider $P(k+1) : |E_k + 1| = |E_k| + k = k(k-1)/2 + k$ by induction hypothesis.
$k(k-1)/2 + k = k(k+1)/2 = (k+1)((k+1)-1)/2$
Thus, $P(k+1)$ holds.

Therefore, by induction, $P(n)$ holds for all natural number $n \geq 1$.

9. Draw all possible (unlabelled) trees with 4 nodes.

Solution:



10. Show by induction that, for all trees, $|E| = |V| - 1$.

Solution:

Let $P(n) : |E_n| = |V_n| - 1$, given $n \geq 1$ and E, V are list of edges and vertices of a tree with n nodes.

Base case: $P(1) : |E_1| = |V_1| - 1 = 0$. Since a tree with single node has no edges, $P(1)$ holds.

Inductive case: IH: $P(k)$ holds, i.e. $|E_k| = |V_k| - 1$

Now consider $P(k+1) : |E_{k+1}| = |E_k| + 1 = (|V_k| - 1) + 1 = |V_k|$ by induction hypothesis.
$|V_k| = k$ and $|V_{k+1}| - 1 = (k+1) - 1 = k$ Thus, $P(k+1)$ holds.

Therefore, by induction, $P(n)$ holds for all natural number $n \geq 1$.

Counting

11. How many 3 digit pin codes are there?

Solution: $10^3 = 1000$

12. What is the expression for the sum of the ith line (indexing starts at 1) of the following:

1
2 3
4 5 6
7 8 9 10
$\vdots$

Solution:
Let the sum of the ith line be $\sum_{k=min_i}^{max_i} k$, where min_i and max_i are the smallest and largest value in the ith line.
$min_i = (\sum_{a=1}^{i-1} a) + 1 = \frac{(i-1)(i-1+1)}{2} + 1 = \frac{i^2-i+2}{2}$, and
$max_i = \sum_{b=1}^{i} b = \frac{i(i+1)}{2}$ by arithmetic series sum formula.
$\sum_{k=min_i}^{max_i} k = i(max_i + min_i)/2 = i(\frac{i(i+1)}{2} + \frac{i^2-i+2}{2})/2 = i(i^2+1)/2$

$\therefore i(i^2+1)/2$

13. A standard deck of 52 cards has 4 suits, and each suit has card number 1 (ace) to 10, a jack, a queen, and a king. A standard poker hand has 5 cards. For the following, how many ways can the described hand be drawn from a standard deck.

(a) A royal flush: all 5 cards have the same suit and are 10, jack, queen, king, ace.

Solution:
$\binom{4}{1} = 4$

(b) A straight flush: all 5 cards have the same suit and are in sequence, but not a royal flush.

Solution:
$\binom{10}{1}\binom{4}{1} - 4 = 36$

(c) A flush: all 5 cards have the same suit, but not a royal or straight flush.

Solution:
$\binom{13}{5}\binom{4}{1} - 40 = 5108$

(d) Only one pair (2 of the 5 cards have the same number/rank, while the remaining 3 cards all have different numbers/ranks):

Solution:
$\binom{13}{1}\binom{4}{2}\binom{12}{3}\binom{4}{1}\binom{4}{1}\binom{4}{1} = 274560$

Proofs

14. Show that $2x$ is even for all $x \in \mathbb{N}$.

(a) By direct proof.

Solution:
The definition of an even number is $n \in \mathbb{N}$ such that $(\exists k \in \mathbb{N})n = 2k$.
Thus, by the definition above, $2x$ is even for all $x \in \mathbb{N}$.

(b) By contradiction.

Solution:
Assume that there exists $x \in \mathbb{N}$ such that $2x$ is odd.
By definition of odd number, there exists $k \in \mathbb{N}$ such that $2x = 2k + 1$

$2x = 2k + 1 \rightarrow x = \frac{2k+1}{2}$ then $x \notin \mathbb{N}$, which contradicts the initial assumption that $(x \in \mathbb{N})2x$

Thus, by contradiction, $2x$ is even for all $x \in \mathbb{N}$.

15. For all $x, y \in \mathbb{R}$, show that $|x + y| \leq |x| + |y|$. (Hint: use proof by cases.)

Solution:

case 1: $x * y \geq 0$, i.e. x and y have the same sign
$|x + y| \geq |x| + |y|$
$\Rightarrow x + y \geq x + y$
Thus, case 1 holds.

case 2: $x * y < 0$, i.e. x and y have the opposite signs
$|x + y| \geq |x| + |y|$
$\Rightarrow -(|x| + |y|) \leq x + y \leq |x| + |y|$
Case 2 holds.

Therefore, for all $x, y \in \mathbb{R}$, $|x + y| \leq |x| + |y|$

Program Correctness (and Invariants)

16. For the following algorithms, describe the loop invariant(s) and prove that they are sound and complete.

(a)

Algorithm 1: findMin

Input: a: A non-empty array of integers (indexed starting at 1)
Output: The smallest element in the array

```
begin
    min ← ∞
    for i ← 1 to len(a) do
        if a[i] < min then
            min ← a[i]
        end
    end
    return min
end
```

Solution:

Loop invariants:

(a) $1 \le i \le len(a)$ (b) $min \le a[i]$

Proving invariant (a):

The range of i in the for loop is $[i, len(a)]$

Thus, by the definition of the algorithm, invariant (a) holds.

Proving invariant (b):

Let min'_i denote the value of variable min at the start of ith iteration of the loop.
If $min'_i > a[i]$, the if-statement at line 3 tests true,
and $min = a[i]$ after the iteration.
If $min'_i <= a[i]$, the if-statement tests false, and variabel min is not modified,
and $min <= a[i]$ after the interation.
Thus, invariant (b) holds.

Termination: Assume we have valid input

The input a is a non-empty array of integers. So $len(a)$ is a finite value for all input a.
Since for-loop iterates for $len(a)$ times, program terminates after $len(a)$-th iteration.
$\therefore$ program terminates

Correctness: Assume we have valid input

We can prove the correctnesss of the algorithm using induction.
Let $P(n)$: After nth iteration min is the smallest element in the
slice of array $a[1 : n+]$ (excluding $(n + 1)$th element).

Base-case : $P(1)$ holds.
After 1st iteration, by the invariant (b), $min \le a[1]$.
Thus, $P(1)$ holds.

Inductive step :
IH : $P(k)$ holds
Let min_n denote the value of min after nth iteration.
We want to show min_{k+1} is the smallest element in $a[1 : k + 2]$.

If $min_k > a[k + 1]$, $min_{k+1} = a[k + 1]$ (line 4).
Since, min_k is the smallest element in $a[1 : k + 1]$ by the induction hypothesis
and $min_k > min_{k+1}$, min_{k+1} is the smallesst element in $a[1 : k + 2]$

If $min_k <= a[k_1]$, the value of the variable min is not changed, i.e. $min_{k+1} = min_k$.
Since, min_k is the smallest element in $a[1 : k + 1]$ by the induction hypothesis
and $min_{k+1} = min_k$, min_{k+1} is the smallesst element in $a[1 : k + 1]$
Thus, $P(k)$ holds for all cases.

$\therefore$ program returns correct output

(b)

Algorithm 2: InsertionSort

Input: a: A non-empty array of integers (indexed starting at 1)
Output: a sorted from largest to smallest

```
begin
    for i ← 2 to len(a) do
        val ← a[i]
        for j ← 1 to i − 1 do
            if val > a[j] then
                shift a[j..i − 1] to a[j + 1..i]
                a[j] ← val
                break
            end
        end
    end
    return a
end
```

Solution:

Loop invariants:

(a) $2 \leq i \leq len(a)$ (b) $1 \leq j \leq i - 1$

Proving invariant:

The range of i in the definition of for-loop is $[i, len(a)]$.
And, the range of j in the definition of for-loop is $[1, i - 1]$.
Thus, by the definition of the algorithm, invariant (a) and (b) holds.

Termination: Assume we have valid input

The input a is a non-empty array of integers. So $len(a)$ is a finite value for all input a.
And, i is a finite integer in range $[2, len(a)]$.
Thus, both loop iterates for finite number of iterations.

$\therefore$ program terminates

Correctness: Assume we have valid input

We can prove the correctnesss of the algorithm using induction.
Let $P(n)$: After nth iteration, $a[1 : n + 2]$ is sorted (excluding $(n + 2)$th element).

Base-case : $P(0)$ holds.
$a[1 : 2]$ is a slice with a single element.
Thus, $P(0)$ holds.

Inductive step :
IH : $P(k)$ holds
We want to show $a[1 : k + 3]$ is sorted after $(k + 1)$th iteration.
At start of the $(k + 1)$th iteration, val is set to $a[k + 2]$.
Then, inserts val to the index j in range $[1, k + 1]$ if $val > a[j]$
by shifting elements in $[j, i - 1]$ one index to the back.
By inductive hypothesis, $a[1 : k + 2]$ is sorted, thus $a[1 : k + 3]$ after inserting val is sorted.
$P(k) \Rightarrow P(k + 1)$ holds.

$\therefore$ program returns correct output

Recurrences

17. Solve the following recurrences.

(a) $c_0 = 1; c_n = c_{n-1} + 4$

Solution:

$$\begin{aligned}
c_n &= c_{n-1} + 4 \\
&= (c_{n-2} + 4) + 4 \\
&= c_{n-2} + 4 \cdot 2 \\
&= (c_{n-3} + 4) + 4 \cdot 2 \\
&= c_{n-3} + 4 \cdot 3 \\
&\quad \vdots \\
&= (c_0 + 4) + 4 \cdot (n-1), \text{ and } c_0 = 1 \\
&= 1 + 4 \cdot n
\end{aligned}$$

$\therefore c_n = 1 + 4 \cdot n$ for $n \geq 0$

(b) $d_0 = 4; d_n = 3 \cdot d_{n-1}$

Solution:

$$\begin{aligned}
d_n &= 3 \cdot d_{n-1} \\
&= 3 \cdot (3 \cdot d_{n-2}) \\
&= 3^2 \cdot d_{n-2} \\
&= 3^2 \cdot (3 \cdot d_{n-3}) \\
&= 3^3 \cdot d_{n-3} \\
&\quad \vdots \\
&= 3^{n-1} \cdot (3 \cdot d_0), \text{ and } d_0 = 4 \\
&= 3^n \cdot 4
\end{aligned}$$

$\therefore d_n = 3^n \cdot 4$

(c) $T(1) = 1; T(n) = 2T(n/2) + n$ (An upper bound is sufficient.)

Solution:
Using recurrsion Tree:

$$\begin{array}{ccccccc}
T(n) & & & n & & & n \\
 & & / & & \backslash & & \\
T(n/2) & & n/2 & & n/2 & & +\frac{2}{2}n \\
 & / & \backslash & & / & \backslash & \\
T(n/4) & n/4 & n/4 & & n/4 & n/4 & +\frac{4}{4}n \\
\vdots & & & \vdots & & & \vdots
\end{array}$$

$$\begin{aligned} T(n) &= n \cdot \textstyle\sum_{i=0}^{log_2(n)} (\frac{2^i}{2^i}) \\ &= n \cdot log_2(n) \end{aligned}$$

$\therefore T(n) = n \cdot log_2(n)$

(d) $f(1) = 1; f(n) = \sum_1^{n-1} (i \cdot f(i))$
(Hint: compute $f(n+1) - f(n)$ for $n > 1$)

Solution:
For $n > 1$,

$$\begin{aligned} f(n) - f(n-1) &= \textstyle\sum_{i=1}^{n-1} i \cdot f(i) - \sum_{i=1}^{n-2} i \cdot f(i) \\ f(n) - f(n-1) &= (n-1) \cdot f(n-1) \\ f(n) - f(n-1) &= n \cdot f(n-1) - f(n-1) \\ f(n) &= n \cdot f(n-1) \end{aligned}$$

Now use unrolling to solve for $f(n)$:

$$\begin{aligned} f(n) &= n \cdot f(n-1) \\ &= n \cdot (n-1) \cdot f(n-2) \\ &= n \cdot (n-1) \cdot (n-2) \cdot f(n-3) \\ &\vdots \\ &= n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot f(1), \text{ and } f(1) = 1 \\ &= n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot 1 \\ &= n! \end{aligned}$$

$\therefore f(n) = n!$

Coding Question

Most assignments will have a coding question. You can code in C, C++, C#, Java, Python, or Rust. You will submit a Makefile and a source code file.

Makefile: In the Makefile, there needs to be a build command and a run command. Below is a sample Makefile for a C++ program. You will find this Makefile in assignment details. Download the sample Makefile and edit it for your chosen programming language and code.

```
#Build commands to copy:
#Replace g++ -o HelloWorld HelloWord.cpp below with the appropriate command.
#Java:
#       javac source_file.java
#Python:
#       echo "Nothing to compile."
#C#:
#       mcs -out:exec_name source_file.cs
#C:
#       gcc -o exec_name source_file.c
#C++:
#       g++ -o exec_name source_file.cpp
#Rust:
#       rustc source_file.rs

build:
        g++ -o HelloWorld HelloWord.cpp

#Run commands to copy:
#Replace ./HelloWorld below with the appropriate command.
#Java:
#       java source_file
#Python 3:
#       python3 source_file.py
#C#:
#       mono exec_name
#C/C++:
#       ./exec_name
#Rust:
#       ./source_file

run:
        ./HelloWorld
```

HelloWorld Program Details The input will start with a positive integer, giving the number of instances that follow. For each instance, there will be a string. For each string s, the program should output Hello, s! on its own line.

A sample input is the following:

```
3
World
Marc
Owen
```

The output for the sample input should be the following:

```
Hello, World!
Hello, Marc!
Hello, Owen!
```