NYU, Tandon School of Engineering
Extended Bridge to CS — Summer 2019

Exam #1

Thursday, August 15, 2019

- This exam has 12 questions, with 100 points total.

- You have two hours.

- **You should submit your answers to the corresponding places in the exam on the NYU Classes system.**

- In total, you should upload 3 '.cpp' files:
  - One '.cpp' file for questions 1-10.
    Write your answer as one long comment (/* … */).
    Name this file 'YourNetID_q1to10.cpp'.
  - One '.cpp' file for question 11, containing your code.
    Name this file 'YourNetID_q11.cpp'.
  - One '.cpp' file for question 12, containing your code.
    Name this file 'YourNetID_q12.cpp'.

- **Write your name, and netID at the head of each file.**

- This is a closed-book exam. However, you are allowed to use
  CLion or Visual-Studio. You should create a new project, and work ONLY in it.
  You may also use two sheets of scratch paper.
  Besides that, no additional resources (of any form) are allowed.

- Calculators are **not** allowed.

- Read every question completely before answering it.
  Note that there are 2 programming problems at the end**.**
  Be sure to allow enough time for these questions

Table 1.5.1: Laws of propositional logic.

Idempotent laws:	$p \vee p \equiv p$	$p \wedge p \equiv p$
Associative laws:	$(p \vee q) \vee r \equiv p \vee (q \vee r)$	$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
Commutative laws:	$p \vee q \equiv q \vee p$	$p \wedge q \equiv q \wedge p$
Distributive laws:	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
Identity laws:	$p \vee F \equiv p$	$p \wedge T \equiv p$
Domination laws:	$p \wedge F \equiv F$	$p \vee T \equiv T$
Double negation law:	$\neg\neg p \equiv p$	
Complement laws:	$p \wedge \neg p \equiv F$ $\neg T \equiv F$	$p \vee \neg p \equiv T$ $\neg F \equiv T$
De Morgan's laws:	$\neg(p \vee q) \equiv \neg p \wedge \neg q$	$\neg(p \wedge q) \equiv \neg p \vee \neg q$
Absorption laws:	$p \vee (p \wedge q) \equiv p$	$p \wedge (p \vee q) \equiv p$
Conditional identities:	$p \rightarrow q \equiv \neg p \vee q$	$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

Table 1.12.1: Rules of inference known to be valid arguments.

Rule of inference	Name
p $p \rightarrow q$ $\therefore q$	Modus ponens
$\neg q$ $p \rightarrow q$ $\therefore \neg p$	Modus tollens
p $\therefore p \vee q$	Addition
$p \wedge q$ $\therefore p$	Simplification

Rule of inference	Name
p q $\therefore p \wedge q$	Conjunction
$p \rightarrow q$ $q \rightarrow r$ $\therefore p \rightarrow r$	Hypothetical syllogism
$p \vee q$ $\neg p$ $\therefore q$	Disjunctive syllogism
$p \vee q$ $\neg p \vee r$ $\therefore q \vee r$	Resolution

Table 1.13.1: Rules of inference for quantified stateme

Rule of Inference	Name
c is an element (arbitrary or particular) ∀x P(x) ∴ P(c)	Universal instantiation
c is an arbitrary element P(c) ∴ ∀x P(x)	Universal generalization
∃x P(x) ∴ (c is a particular element) ∧ P(c)	Existential instantiation*
c is an element (arbitrary or particular) P(c) ∴ ∃x P(x)	Existential generalization

Table 3.6.1: Set identities.

Name	Identities	
Idempotent laws	A ∪ A = A	A ∩ A = A
Associative laws	(A ∪ B) ∪ C = A ∪ (B ∪ C)	(A ∩ B) ∩ C = A ∩ (B ∩ C)
Commutative laws	A ∪ B = B ∪ A	A ∩ B = B ∩ A
Distributive laws	A ∪ (B ∩ C) = (A ∪ B) ∩ (A ∪ C)	A ∩ (B ∪ C) = (A ∩ B) ∪ (A ∩ C)
Identity laws	A ∪ ∅ = A	A ∩ U = A
Domination laws	A ∩ ∅ = ∅	A ∪ U = U
Double Complement law	$\overline{\overline{A}} = A$	
Complement laws	$A \cap \overline{A} = \varnothing$ $\overline{U} = \varnothing$	$A \cup \overline{A} = U$ $\overline{\varnothing} = U$
De Morgan's laws	$\overline{A \cup B} = \overline{A} \cap \overline{B}$	$\overline{A \cap B} = \overline{A} \cup \overline{B}$
Absorption laws	A ∪ (A ∩ B) = A	A ∩ (A ∪ B) = A

Part I – Theoretical:

- ***You don't need to justify your answers to the questions in this part.***
- *For all questions in this part of the exam (questions 1-10), you should submit a* ***single*** *'.cpp' file. Write your answers as one long comment (`/* … */`).*
  Name this file 'YourNetID_q1to10.cpp'.

Question 1 (5 points)

a. Convert the decimal number $(168)_{10}$ to its base-2 representation.
b. Convert the 8-bits two's complement number $(10101000)_{\text{8-bit two's complement}}$ to its decimal representation.

Question 2 (5 points)

Select the statement that is equivalent to:
"It is not true that the patient has high blood pressure or influenza."
a. The patient has high blood pressure or has influenza.
b. The patient does not have high blood pressure and does not have influenza.
c. The patient does not have high blood pressure or does not have influenza.
d. The patient has high blood pressure and has influenza.

Question 3 (5 points)

The domain for variable x is the set {Ann, Ben, Cam, Dave}.
The table below gives the values of predicates P and Q for every element in the domain.

	P(x)	Q(x)
Ann	F	F
Ben	T	F
Cam	T	T
Dave	T	T

Select the statement that is **true**.
a. $\forall x\ (Q(x) \rightarrow P(x))$
b. $\forall x\ (P(x) \rightarrow Q(x))$
c. $\forall x\ (P(x) \wedge Q(x))$
d. $\forall x\ (P(x) \vee Q(x))$

Question 4 (5 points)

The domain of discourse for x and y is the set of employees at a company. *Miguel* is one of the employees at the company.

Define the predicates:

V(x): x is a manager

M(x, y): x earns more than y

Select the logical expression that is equivalent to:

“Everyone who earns more than Miguel is a manager.”

a. $\forall x\ (M(x, Miguel)\ \rightarrow\ \neg V(x))$

b. $\forall x\ (M(x, Miguel)\ \wedge\ \neg V(x))$

c. $\neg\exists x\ (M(V(x), Miguel)$

d. $\neg\exists x\ (M(x, Miguel)\ \wedge\ \neg V(x))$

Question 5 (5 points)

The domain for variable x is the set of all integers.

Select the correct rules to replace (?) in lines 3 and 4 of the proof segment below:

1.	∀x $(P(x) \wedge Q(x))$	Hypothesis
2.	3 is an integer	Hypothesis
3.	$P(3) \wedge Q(3)$	(?)
4.	$P(3)$	(?)

a. Universal generalization; Simplification

b. Universal generalization; Conjunction

c. Universal instantiation; Simplification

d. Universal instantiation; Conjunction

Question 6 (5 points)

Theorem: There is no smallest positive rational number.

A proof by contradiction of the theorem starts by assuming which fact?

a. Let r be an arbitrary positive rational number.

b. Let r be the smallest rational number.

c. Let r be the smallest positive real number.

d. Let r be the smallest positive rational number.

Question 7 (5 points)

Select the set that is equivalent to $A - (A \cup B)$.

a. A
b. B
c. $\emptyset$
d. $A - B$

Question 8 (10 points)

$A = \{1, 2, \{3, 4\}, \{\}\}$.

For each of the following statements, state if they are **true or false** (no need to explain your choice).

a. $1 \in A$
b. $1 \subseteq A$
c. $\{3\} \in A$
d. $\{3\} \subseteq A$
e. $\{1, 2\} \in A$
f. $\{1, 2\} \subseteq A$
g. $\{3, 4\} \subseteq A$
h. $\{\{3, 4\}\} \subseteq A$
i. $\emptyset \in A$
j. $\emptyset \subseteq A$

Question 9 (5 points)

Consider the following function:

$f: \{0,1\}^3 \to \{0,1\}^5$. f(x) is obtained from x by adding 0 to its start and 1 to its end.
For example, f(101) = 01011.
Select the correct description of the function f.

a. One-to-one and onto
b. One-to-one but not onto
c. Onto but not one-to-one
d. Neither one-to-one nor onto

Question 10 (5 points)

Let A={1, 2, 3}.

The function $f: A \times A \to Z$ is defined as: for every $(x, y) \in A \times A$, f((x, y)) = $x^2 - y$.
For example, $f((2, 3)) = 1$ (Since: $2^2 - 3$ is 1),

Find the range of f

Part II – Coding:

- *For **each** question in this part (questions 11-12), you should submit a '.cpp' file, containing your code.*
- *Pay special attention to the style of your code. Indent your code correctly, choose meaningful names for your variables, define constants where needed, choose most suitable control statements, etc.*
- *In all questions, you may assume that the user enters inputs as they are asked.*
  For example, if the program expects a positive integer, you may assume that user will enter positive integers.
- *No need to document your code. However, you may add comments if you think they are needed for clarity.*

Question 11 (20 points)

Write a program that reads an integer greater or equal to 2, n, and prints a shape of a n-line hollow inverted pyramid of stars.

Your program should interact with the user **exactly** as it shows in the following two executions:

Execution example 1:

```
Please enter an integer, greater or equal to 2:
5
*********
 *     *
  *   *
   * *
    *
```

Execution example 2:

```
Please enter an integer, greater or equal to 2:
3
*****
 * *
  *
```

Question 12 (25 points)

Consider the following definition:
A positive integer *num* is called a *factorion* if it equals to the sum of the factorials of its digits.

For example, 145 is a factorion because 1! + 4! + 5! = 1 + 24 + 120 = 145.

Write a program that asks the user to enter a positive integer and reports if that number is a factorion or not.

Reminder: the factorial of a positive integer *n*, denoted by *n!*, is the product of all positive integers less than or equal to *n*: $n! = n \times (n-1) \times (n-2) \times \ldots \times 2 \times 1$.
For example, $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$
Also, the value of 0! is defined as 1

Your program should interact with the user **exactly** as demonstrated in the following two executions:

Execution example 1:

```
Please enter a positive integer:
145
145 is a factorion
```

Execution example 2:

```
Please enter a positive integer:
87
87 is not a factorion
```