NYU, Tandon School of Engineering
Extended Bridge to CS — Winter 2021

Exam #1

Thursday, February 11, 2021

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

- You have **two hours**.

- **You should submit your answers in the Gradescope platform** (not on NYU Classes).

- **It is your responsibility to take the time for the exam** (You may use a physical timer, or an online timer: https://vclock.com/set-timer-for-2-hours/).
  Make sure to upload the files with your answers to gradescope BEFORE the time is up, while still being monitored by ProctorU.
  We will not accept any late submissions.

- 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.
  - Two sheets of scratch paper.
  - Scientific Calculator

  Besides that, no additional resources (of any form) are allowed.

- You are not allowed to use C++ syntactic features that were not covered in the Bridge program so far.

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

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 $\therefore p \lor q$	Addition
$p \land q$ $\therefore p$	Simplification

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

Table 1.13.1: Rules of inference for quantified stateme

Rule of Inference	Name
c is an element (arbitrary or particular) $\underline{\forall x\ P(x)}$ $\therefore$ P(c)	Universal instantiation
c is an arbitrary element $\underline{P(c)\qquad}$ $\therefore \forall x\ P(x)$	Universal generalization
$\underline{\exists x\ P(x)}$ $\therefore$ (c is a particular element) $\wedge$ P(c)	Existential instantiation*
c is an element (arbitrary or particular) $\underline{P(c)\qquad}$ $\therefore \exists x\ P(x)$	Existential generalization

Table 3.6.1: Set identities.

Name	Identities	
Idempotent laws	$A \cup A = A$	$A \cap A = A$
Associative laws	$(A \cup B) \cup C = A \cup (B \cup C)$	$(A \cap B) \cap C = A \cap (B \cap C)$
Commutative laws	$A \cup B = B \cup A$	$A \cap B = B \cap A$
Distributive laws	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
Identity laws	$A \cup \varnothing = A$	$A \cap U = A$
Domination laws	$A \cap \varnothing = \varnothing$	$A \cup U = U$
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	$A \cup (A \cap B) = A$	$A \cap (A \cup B) = A$

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 $(148)_{10}$ to its **base-4** representation.
b. Convert the 8-bits two's complement number $(10111100)_{\text{8-bit two's complement}}$ to its decimal representation.

Question 2 (5 points)

Select the proposition that is logically equivalent to $\neg(\neg p \rightarrow q)$.

a. $p \wedge q$
b. $p \wedge \neg q$
c. $\neg p \wedge q$
d. $\neg p \wedge \neg q$
e. None of the above

Question 3 (5 points)

The domain for variables x and y is the set {1, 2, 3}.
The table below gives the values of P(x,y) for every pair of elements from the domain.
For example, P(2, 3) = F because the value in row 2, column 3, is F.

P	1	2	3
1	T	T	F
2	T	T	F
3	F	T	F

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

a. $\exists x\, \forall y\, P(x, y)$
b. $\forall x\, \exists y\, P(x, y)$
c. $\exists y\, \forall x\, P(x, y)$
d. $\forall y\, \exists x\, P(x, y)$

Question 4 (5 points)

The domain of discourse is the set of employees at a 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:
"There is an employee that is not a manager, which earns more than at least one manager"

a. $\neg\forall x\ (V(x) \land \exists y\ (V(y) \rightarrow M(x,y)))$
b. $\exists x\ (\neg V(x) \land \exists y\ (V(y) \land M(x,y)))$
c. $\neg\forall x\ (V(x) \rightarrow \exists y\ (V(y) \land M(x,y)))$
d. $\exists x\ (\neg V(x) \land \forall y\ (V(y) \rightarrow M(x,y)))$

Question 5 (5 points)

$A = \{x \in Z: x\ is\ odd\}$
$B = \{x \in Z: x\ is\ a\ power\ of\ 2\}$
$D = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

Select the set corresponding to $D - (A \cup B)$.

a. $\{3, 5, 6, 7, 9, 10\}$
b. $\{1, 2, 4, 8\}$
c. $\{6, 10\}$
d. $\{1, 2, 3, 4, 5, 7, 8, 9\}$

Question 6 (5 points)

The domain for variable x is the set of all integers.
Select the correct rules to replace (?) in lines 2 and 3 of the proof segment below:

1.	$\neg P(3)$	Hypothesis
2.	$\neg P(3) \lor Q(3)$	(?)
3.	$\exists \mathrm{x}\ (\neg P(x) \lor Q(x))$	(?)

a. Addition; Existential instantiation
b. Disjunctive syllogism; Existential instantiation
c. Addition; Existential generalization
d. Disjunctive syllogism; Existential generalization

Question 7 (5 points)

Theorem: The average of any two real numbers is greater than or equal to at least one of the two numbers.

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

a. For every two real numbers, x and y, $\left(\frac{x+y}{2} < x\right) \wedge \left(\frac{x+y}{2} < y\right)$.

b. For every two real numbers, x and y, $\left(\frac{x+y}{2} < x\right) \vee \left(\frac{x+y}{2} < y\right)$.

c. There exists two real numbers, x and y, such that $\left(\frac{x+y}{2} < x\right) \vee \left(\frac{x+y}{2} < y\right)$.

d. There exists two real numbers, x and y, such that $\left(\frac{x+y}{2} < x\right) \wedge \left(\frac{x+y}{2} < y\right)$.

e. None of the above.

Question 8 (10 points)

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

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. $\{1\} \in A$
d. $\{1\} \subseteq A$
e. $\{2\} \in P(A)$
f. $\{2\} \subseteq P(A)$
g. $\emptyset \in A$
h. $\emptyset \subseteq A$
i. $\emptyset \in P(A)$
j. $\emptyset \subseteq P(A)$

Question 9 (5 points)

$f: \mathrm{N} \to \mathrm{N}$. $f(n) = n + 1$.

Select the correct description of the function f.

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

Question 10 (5 points)

The function $f: \{0,1\}^2 \to \{0,1\}^4$ is defined as:

For every $s \in \{0,1\}^2$, f(s) = 0s1. That is, f **adds** 0 to its beginning and 1 to its end.

For example, f(00)=0001

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 a lower-case letter $inLetter$, and prints a shape of an isosceles triangle. The first line of the triangle should have the letter a, the second line of the triangle should have the letter b, the third line of the triangle should have the letter c, ..., the last line of the triangle should have the letter $inLetter$.
In addition, the casing of the letters should alternate between lines. That is, in the first line, the 'a' should be lower case. In the second line, all 'b's should be upper case. In the third line, all the 'c's should be lower case, etc.

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

Execution example 1:
```
Please enter a lower-case letter:
e
    a
   BBB
  ccccc
 DDDDDDD
eeeeeeeee
```

Execution example 2:
```
Please enter a lower-case letter:
d
   a
  BBB
 ccccc
DDDDDDD
```

Question 12 (25 points)
In this question, we will calculate the total distance a snail travels over a sequence of days.

Write a program that asks the user to enter the number of inches a snail traveled in each day. The numbers should be entered in separate lines, and end with -1 (to indicate the end of the sequence).
After reading the numbers, your program should print the total distance the snail traveled. This distance should be printed in a (miles, yards, inches) format

Note:
You may use the following facts:

1 foot = 12 inches
1 yard = 3 feet

Your program should interact with the user **exactly** as demonstrated bellow.

Execution example:
```
Enter the number of inches the snail traveled in day #1,
or type -1 if they reached their destination:
132
Enter the number of inches the snail traveled in day #2,
or type -1 if they reached their destination:
440
Enter the number of inches the snail traveled in day #3,
or type -1 if they reached their destination:
265
Enter the number of inches the snail traveled in day #4,
or type -1 if they reached their destination:
7
Enter the number of inches the snail traveled in day #5,
or type -1 if they reached their destination:
-1
In 4 days, the snail traveled 23 yards, 1 feet and 4 inches.
```