NYU, Tandon School of Engineering
CS Bridge — Spring 2023

# Exam 1

## Thursday, May 04, 2023

- This exam has 14 questions, with 100 points total.

- You have **two hours**.

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

- **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 1-12.
    Write your answer as one long comment (/* … */).
    Name this file 'YourNetID_q1to12.cpp'.
  - One '.cpp' file for question 13, containing your code.
    Name this file 'YourNetID_q13.cpp'.
  - One '.cpp' file for question 14, containing your code.
    Name this file 'YourNetID_q14.cpp'.

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

- This is a closed-book exam. However, you are allowed to use:
  - Visual Studio Code (VSCode) or Visual-Studio or Xcode or CLion. You should create a new project and work ONLY in it.
  - Two sheets of scratch paper.

  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: | $p \land \neg p \equiv F$<br>$\neg T \equiv F$ | $p \lor \neg p \equiv T$<br>$\neg F \equiv T$ |
| 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: | $p \to q \equiv \neg p \lor q$ | $p \leftrightarrow q \equiv (p \to q) \land (q \to p)$ |

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

| Rule of inference | Name |
|---|---|
| $p$<br>$p \to q$<br>$\therefore q$ | Modus ponens |
| $\neg q$<br>$p \to q$<br>$\therefore \neg p$ | Modus tollens |
| $p$<br>$\therefore p \lor q$ | Addition |
| $p \land q$<br>$\therefore p$ | Simplification |

| Rule of inference | Name |
|---|---|
| $p$<br>$q$<br>$\therefore p \land q$ | Conjunction |
| $p \to q$<br>$q \to r$<br>$\therefore p \to r$ | Hypothetical syllogism |
| $p \lor q$<br>$\neg p$<br>$\therefore q$ | Disjunctive syllogism |
| $p \lor q$<br>$\neg p \lor r$<br>$\therefore q \lor r$ | Resolution |

Table 1.13.1: Rules of inference for quantified stateme

| Rule of Inference | Name |
|---|---|
| c is an element (arbitrary or particular)<br>∀x P(x)<br>∴ P(c) | Universal instantiation |
| c is an arbitrary element<br>P(c)<br>∴ ∀x P(x) | Universal generalization |
| ∃x P(x)<br>∴ (c is a particular element) ∧ P(c) | Existential instantiation* |
| c is an element (arbitrary or particular)<br>P(c)<br>∴ ∃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 | $A \cap \overline{A} = \varnothing$<br>$\overline{U} = \varnothing$ | $A \cup \overline{A} = U$<br>$\overline{\varnothing} = U$ |
| 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 multiple choice questions, there could be more than one answer.***
- *For all questions in this part of the exam (questions 1-12), you should submit a* ***single*** *'.cpp' file. Write your answers as one long comment (/* … */).*
  *Name this file 'YourNetID_q1to12.cpp'.*

**Question 1 (8 points)**

a. Convert the decimal number $(2819)_{10}$ to its **base-7** representation.
b. Convert the 8-bits two's complement number $(11100101)_{\text{8-bit two's complement}}$ to its decimal representation.

**Question 2 (4 points)**

Select the propositions that are logically equivalent to $(q \rightarrow \neg p)$.

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

**Question 3 (5 points)**

The domain of the variable x consists of all people in the world, the domain of the variable y consists of set of all movies. Define the predicates:
S(x, y): x saw y
L(x, y): x liked y.

Select the logical expression that is equivalent to: "No one liked every movie he has seen."

a. $\exists x \forall y\ (S(x,y)\ \rightarrow\ L(x,y))$
b. $\neg\forall x \exists y\ (S(x,y)\ \rightarrow\ L(x,y))$
c. $\neg\exists x \forall y\ (S(x,y)\ \rightarrow\ L(x,y))$
d. $\neg\exists x \forall y\ (S(x,y)\ \wedge\ L(x,y))$
e. None of the above

**Question 4 (5 points)**

Suppose you want to prove a theorem of the form "**if $\neg q$ then $\neg p$**". If you give a proof by contraposition, what do you assume and what do you prove?
a. Assume $\neg p$ is true, prove that ¬q is true.
b. Assume $p$ is true, prove that $q$ is true.
c. Assume $\neg q$ is true, prove that ¬p is true.
d. Assume $(\neg p \vee q)$ is true, prove that $q$ is true.
e. None of the above

**Question 5 (5 points)**

Select the logical expressions that is equivalent to: $\forall y \exists x \exists z\, \neg(P(x,y,z) \vee \neg Q(x,y))$

a. $\forall y \exists x\, \neg \forall z\, (P(x,y,z) \vee Q(x,y))$
b. $\neg \exists y \forall x \forall z\, (\neg P(x,y,z) \wedge Q(x,y))$
c. $\forall y \exists x \exists z\, (\neg P(x,y,z) \vee Q(x,y)$
d. $\exists y \forall x \forall z\, (P(x,y,z) \wedge \neg Q(x,y))$
e. None of the above

**Question 6 (5 points)**

Determine whether the following set is the power set of some set. If the following set is a power set, give the set of which it is a power set.

$$\{\emptyset, \{\emptyset\}, \{\boldsymbol{a}\}, \{\boldsymbol{b}\}, \{\emptyset, \boldsymbol{a}\}, \{\emptyset, \boldsymbol{b}\}, \{\mathbf{a}, \boldsymbol{b}\}, \{\emptyset, \boldsymbol{a}, \boldsymbol{b}\}\}$$

a. No, this set is not a power set of any set.
b. Yes, and the set is $\{\emptyset, \boldsymbol{a}\}$
c. Yes, and the set is $\{\emptyset, \boldsymbol{a}, \boldsymbol{b}\}$
d. Yes, and the set is $\{\boldsymbol{a}, \boldsymbol{b}\}$
e. Can't be determined

**Question 7 (10 points)**

$A = \{a, b, c, d, \{b\}, \{d\}, \{a, b\}\}$.

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

a. $c \in A$
b. $\{d\} \subseteq A$
c. $\{a, b, c\} \in A$
d. $\{a, b\} \subseteq A$
e. $\{d\} \in A$
f. $(a, \{a, b\}) \in A \times A$
g. $\{a, b, c,\ \{b\}\} \in P(A)$
h. $\{a, b, c,\ \{b\}\} \subseteq A$
i. $\emptyset \in A$
j. $\{\emptyset\} \subseteq P(A)$

**Question 8 (5 points)**

Select the set that is equivalent to $\overline{A} \cap (A \cup \overline{B})$.

a. $\emptyset$
b. $U$
c. $\overline{A} \cup \overline{B}$
d. $\overline{A} \cap \overline{B}$
e. None of the above

**Question 9 (5 points)**
Let M be defined to be the set $\{1, 2, 3, 4\}$.

Let $f$ be a function: $f: \mathrm{P}(M) \rightarrow P(M)$, defined as follows:

for $X \subseteq M$, $f(X) = M - X$.

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
e. None of the above

**Question 10 (5 points)**
The domain and target set of functions f and g are **Z**. The functions are defined as: $f(x) = 5x + 7$ and $g(x) = 2x^2 + 1$

An explicit formula for the function: g o f(x) will be
a. $50x^2 + 140x + 98$
b. $10x^2 + 12$
c. $50x^2 + 140x + 99$
d. $50x^2 + 70x + 99$
e. None of the above

**Question 11 (5 points)**

Let f be the function from the set of all real numbers to the set of all real numbers with $f(x) = 2x^2 + 1$. Select the statements that are **true.**

a. $f^{-1}(x)$ = sqrt((x-1)/2).
b. f(x) is not invertible.
c. f(x) is both one to one and onto function.
d. f(x) is one to one function but not onto function
e. None of the above

**Question 12 (3 points)**
If it snows today, the university will be closed. The university will not be closed today. Therefore, it did not snow today.

What is the rule of inference being used in the above statement:

a. Resolution
b. Modus Tollens
c. Modus Ponens
d. Hypothetical Syllogism
e. None of the above

## ***Part II – Coding:***

- *For **each** question in this part (questions 13-14), 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 13 (17 points)**

Write a C++ program that reads a positive integer, $n$, and prints a shape of (4n-3) lines consisting of asterisks (*) and spaces as follows:

1st line: print (2n-2) spaces and then print 1 asterisk
2nd line: print (2n-3) spaces and then print 2 asterisks
3rd line: print (2n-4) spaces and then print 3 asterisks
4th line: print (2n-5) spaces and then print 4 asterisks
…
…
…
(2n-1)th line: print 0 space and then print (2n - 1) asterisks
(2n)th line: print 1 space and then print (2n-2) asterisks
(2n+1)th line: print 2 spaces and then print (2n-3) asterisks
(2n+2)th line: print 3 spaces and then print (2n-4) asterisks
…
…
…

(4n-4)th line: print (2n-3) spaces and then print 2 asterisks
(4n-3)th line: print (2n-2) spaces and then print 1 asterisk

Your program should interact with the user **exactly** as demonstrated in the following two executions (color is used just for the illustration purpose only):

**<u>Execution example 1</u>:**

```
Please enter a positive integer:
3
    *
   **
  ***
 ****
*****
 ****
  ***
   **
    *
```

**<u>Execution example 2</u>:**

```
Please enter a positive integer:
5
        *
       **
      ***
     ****
    *****
   ******
  *******
 ********
*********
 ********
  *******
   ******
    *****
     ****
      ***
       **
        *
```

**<u>Question 14 (18 points)</u>**

A sequence of positive numbers has been given. Each of these positive numbers will have at least 2 digit and at most 8 digits. The first digit of these numbers will not be 0 (Zero). Suppose we define different number groups as follows:

**Numbers Group 1**: Total sum of first digit and last digit in each number of this group should be less than 5.
**Numbers Group 2**: Total sum of first digit and last digit in each number of this group should be greater or equal to 5 and less than 10.
**Numbers Group 3**: Total sum of first digit and last digit in each number of this group should be greater or equal to 10 and less than 15.
**Numbers Group 4**: Total sum of first digit and last digit in each number of this group should be greater or equal to 15.

Write a C++ program that reads from the user a sequence of numbers (positive numbers with at least 2-digit and at most 8 digits) and prints the following statistics.

Total count of numbers in the Numbers Group 1:
Total count of numbers in the Numbers Group 2:
Total count of numbers in the Numbers Group 3:
Total count of numbers in the Numbers Group 4:

**<u>Implementation requirement</u>:**

a. **The user should enter their numbers, each one in a separate line, and type -1 to indicate the end of the input.**
b. You are not allowed to use C++ syntactic features that were not covered in the Bridge program so far.
c. You are not allowed to use any **cmath** or **math.h** library function for this program. You have to calculate without using any library function.
d. The first digit of these numbers will not be 0 (Zero).

Your program should interact with the user **exactly** the same way, as demonstrated in the following two executions (color is used just for the illustration purpose only):

**Execution example 1**:

```
Please enter a sequence of numbers (with at least 2-digit and at most 8-
digits), each one in a separate line. End your sequence by typing -1:
123
986
445
2001
324
87123457
90001
12
61
98762345
12345
29
787
111111
161819
340000
9999999
-1
Total count of numbers in the Numbers Group 1: 5
Total count of numbers in the Numbers Group 2: 4
Total count of numbers in the Numbers Group 3: 5
Total count of numbers in the Numbers Group 4: 3
```

**Execution example 2:**

```
Please enter a sequence of numbers (with at least 2-digit and at most 8-
digits), each one in a separate line. End your sequence by typing -1:

120001
123456
83726261
306
98
223
2000001
13
9873
297
21343456
98798
12346
98654
11234
59
99
9999
5558
1293
123
-1
Total count of numbers in the Numbers Group 1: 5
Total count of numbers in the Numbers Group 2: 8
Total count of numbers in the Numbers Group 3: 4
Total count of numbers in the Numbers Group 4: 4
```