Exam 1

Thursday, August 11, 2022

- This exam has 13 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 1-11.
    Write your answer as one long comment (/* … */).
    Name this file 'YourNetID_q1to11.cpp'.
  - One '.cpp' file for question 12, containing your code.
    Name this file 'YourNetID_q12.cpp'.
  - One '.cpp' file for question 13, containing your code.
    Name this file 'YourNetID_q13.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. 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:	$p \vee p \equiv p$	$p \wedge p \equiv p$
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:	$p \to q \equiv \neg p \vee q$	$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$

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

Rule of inference	Name
p $p \to q$ $\therefore q$	Modus ponens
$\neg q$ $p \to q$ $\therefore \neg p$	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
$p \to q$ $q \to r$ $\therefore p \to r$	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 statemer

Rule of Inference	Name
c is an element (arbitrary or particular) $\forall x\, P(x)$ $\therefore P(c)$	Universal instantiation
c is an arbitrary element $P(c)$ $\therefore \forall x\, P(x)$	Universal generalization
$\exists x\, P(x)$ $\therefore$ (c is a particular element) $\wedge\, P(c)$	Existential instantiation*
c is an element (arbitrary or particular) $P(c)$ $\therefore \exists x\, P(x)$	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	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	$A \cap (B \cup C) = (A \cup B) \cap (A \cup C)$
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 1-11), you should submit a* ***single*** *'.cpp' file. Write your answers as one long comment (`/* … */`).*
  Name this file 'YourNetID_q1to11.cpp'.

Question 1 (8 points)

a. Convert the decimal number $(3478)_{10}$ to its **base-7** representation.
b. Convert the 8-bits two's complement number $(11110001)_{\text{8-bit two's complement}}$ to its decimal representation.

Question 2 (4 points)

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

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

Question 3 (5 points)

The domain of the variable x consists of all the students in this university, the domain of the variable y consists of all the courses offered by all the departments of this university, and the domain of the variable z consists of all the departments of this university. Define the predicates:
T(x, y) : student x has taken course y.
O(y, z): course y is offered by department z.

Select the logical expression that is equivalent to:
"There is a student in this university who has taken every course offered by one of the departments in this school."

a. $\exists x \forall z \exists y\ (T(x,y) \rightarrow O(y,z))$
b. $\exists x \exists z \forall y\ (O(y,z) \rightarrow T(x,y))$
c. $\exists x \forall z \forall y\ (O(y,z) \rightarrow T(x,y))$
d. $\exists x \exists z \forall y\ (O(y,z) \wedge T(x,y))$
e. None of the above

Question 4 (5 points)

Given any 40 people, show that at least four of them were born in the same month of the year.

A proof by contradiction of the above statement starts by assuming which fact?

a. Suppose that at least three of the 40 people were born in the same month of the year.
b. Suppose that at most four of the 40 people were born in the same month of the year.
c. Suppose that at most three of the 40 people were born in the same month of the year.
d. Suppose that exactly four of the 40 people were born in the same month of the year.
e. None of the above

Question 5 (5 points)

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

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

Question 6 (4 points)

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

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

Question 7 (10 points)

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

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

a. $a \in A$
b. $b \subseteq A$
c. $\{a, b, d\} \in A$
d. $\{a, b, c\} \subseteq A$
e. $\{d\} \in A$
f. $\{d\} \subseteq A$
g. $\{a, b, c,\ \{b\}\} \in A$
h. $\{a, b, c,\ \{b\}\} \subseteq A$
i. $\emptyset \in A$
j. $\emptyset \subseteq P\ (A)$
k. $|\emptyset| = 1$
l. $|A| = 7$

Question 8 (4 points)

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

a. $\emptyset$
b. U
c. $\overline{A} \cup B$
d. $\overline{A} \cap B$
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 ⊆ 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

Question 10 (5 points)

The domain and target set of functions f and g are **Z**. The functions are defined as: f(x) = 2x + 3 and g(x) = $5x^2 + 7$

An explicit formula for the function: g o f(x) will be

a. $50x^2 + 140x + 101$
b. $50x^2 + 140X+98$
c. $20x^2 + 60X+52$
d. $20x^2 + 60X+45$
e. None of the above

Question 11 (5 points)

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

a. $f^{-1}(x) = \text{sqrt}(x)$
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

Part II – Coding:

- *For **each** question in this part (questions 12-13), 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 12 (20 points)

Write a C++ program that reads a positive integer, n, and prints a shape of (2*n + 1) lines consisting of asterisks (*) and spaces as follows:

1st line: print n asterisks, then no/zero spaces and print n asterisks
2nd line: print 1 asterisk, then print (2*n - 2) spaces and then 1 asterisk
3rd line: print 2 asterisks, then (2*n - 4) spaces and then 2 asterisks
4th line: print 3 asterisks, then (2*n - 6) spaces and then 3 asterisks
…
…
…
(n+1)th line: print n asterisks, then no/zero spaces and print n asterisks
(n+2)th line: print (n - 1) asterisks, then 2 spaces and print (n - 1) asterisks
(n+3)th line: print (n - 2) asterisks, then 4 spaces and print (n - 2) asterisks
…
…
…
(2*n-1)th line: print 2 asterisks, then (2*n - 4) spaces and then 2 asterisks
(2*n)th line: print 1 asterisk, then print (2*n - 2) spaces and then 1 asterisk
(2n+1)th line: print n asterisks, then no/zero spaces and print n asterisks

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

Execution example 1:

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

Execution example 2:

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

Execution example 3:

```
Please enter a positive integer:
6
************
*          *
**        **
***      ***
****    ****
*****  *****
************
*****  *****
****    ****
***      ***
**        **
*          *
************
```

Execution example 4:

```
Please enter a positive integer:
8
****************
*              *
**            **
***          ***
****        ****
*****      *****
******    ******
*******  *******
****************
*******  *******
******    ******
*****      *****
****        ****
***          ***
**            **
*              *
****************
```

Question 13 (20 points)

A sequence of positive numbers has been given. Each of these positive numbers will have at least 3 digits and at most 7 digits. The first digit of these numbers will not be 0 (Zero) and odd digits are (1, 3, 5, 7, 9) and even digits are (0, 2, 4, 6, 8). Suppose we define different number groups as follows:

3-digits Numbers Group: Numbers that have exactly 3-digits.
4-digits Numbers Group: Numbers that have exactly 4-digits.
5-digits Numbers Group: Numbers that have exactly 5-digits.
6-digits Numbers Group: Numbers that have exactly 6-digits.
7-digits Numbers Group: Numbers that have exactly 7-digits.
More Odd-digits Group: Numbers that have more odd-digits than even-digits.
Equal-digits Group: Numbers that have equal number of odd-digits and even-digits.

Write a C++ program that reads from the user a sequence of numbers (positive numbers with at least 3-digits and at most 7 digits) and prints the following statistics.

Total count of numbers in the 3-digits Numbers group:
Total count of numbers in the 4-digits Numbers group:
Total count of numbers in the 5-digits Numbers group:
Total count of numbers in the 6-digits Numbers group:
Total count of numbers in the 7-digits Numbers group:
Total count of numbers in the More Odd-digits group
Total count of numbers in the Equal-digits group:

Implementation requirement:

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) and odd digits are (1, 3, 5, 7, 9) and even digits are (0, 2, 4, 6, 8).

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 3-digits and at most 7-digits), each one in a separate line. End your sequence by typing -1:
154
59987
509
1002
3206
8712345
8123
936767
56123
1988
2601
3998873
2103
420
503
60000
200001
987456
-1
Total count of numbers in the 3-digits Numbers Group: 4
Total count of numbers in the 4-digits Numbers Group: 6
Total count of numbers in the 5-digits Numbers Group: 3
Total count of numbers in the 6-digits Numbers Group: 3
Total count of numbers in the 7-digits Numbers Group: 2
Total count of numbers in the More Odd-digits Group: 8
Total count of numbers in the Equal-digits Group: 4
```

Execution example 2:

```
Please enter a sequence of numbers in between 1 and 100, each one in a
separate line. End your sequence by typing -1:
3335
67123
589
400
60467
3912345
810098
1234451
2398123
14234
1798
4608
1012
2998
35123
46987
251234
509876
361234
259877
21434
11234
698776
78123
1212234
631
990
2409
231245
35124
599877
411234
361234
269872
15123
2418764
2612098
84125
108
4513
3412
-1
Total count of numbers in the 3-digits Numbers Group: 5
Total count of numbers in the 4-digits Numbers Group: 8
Total count of numbers in the 5-digits Numbers Group: 11
Total count of numbers in the 6-digits Numbers Group: 11
Total count of numbers in the 7-digits Numbers Group: 6
Total count of numbers in the More Odd-digits Group: 17
Total count of numbers in the Equal-digits Group: 11
```