MEMORIAL
UNIVERSITY

Faculty of Engineering and Applied Science

St. John's, NL Canada A1B 3X5
Tel: 709 864 8812 Fax: 709 864 4042
https://www.mun.ca/engineering
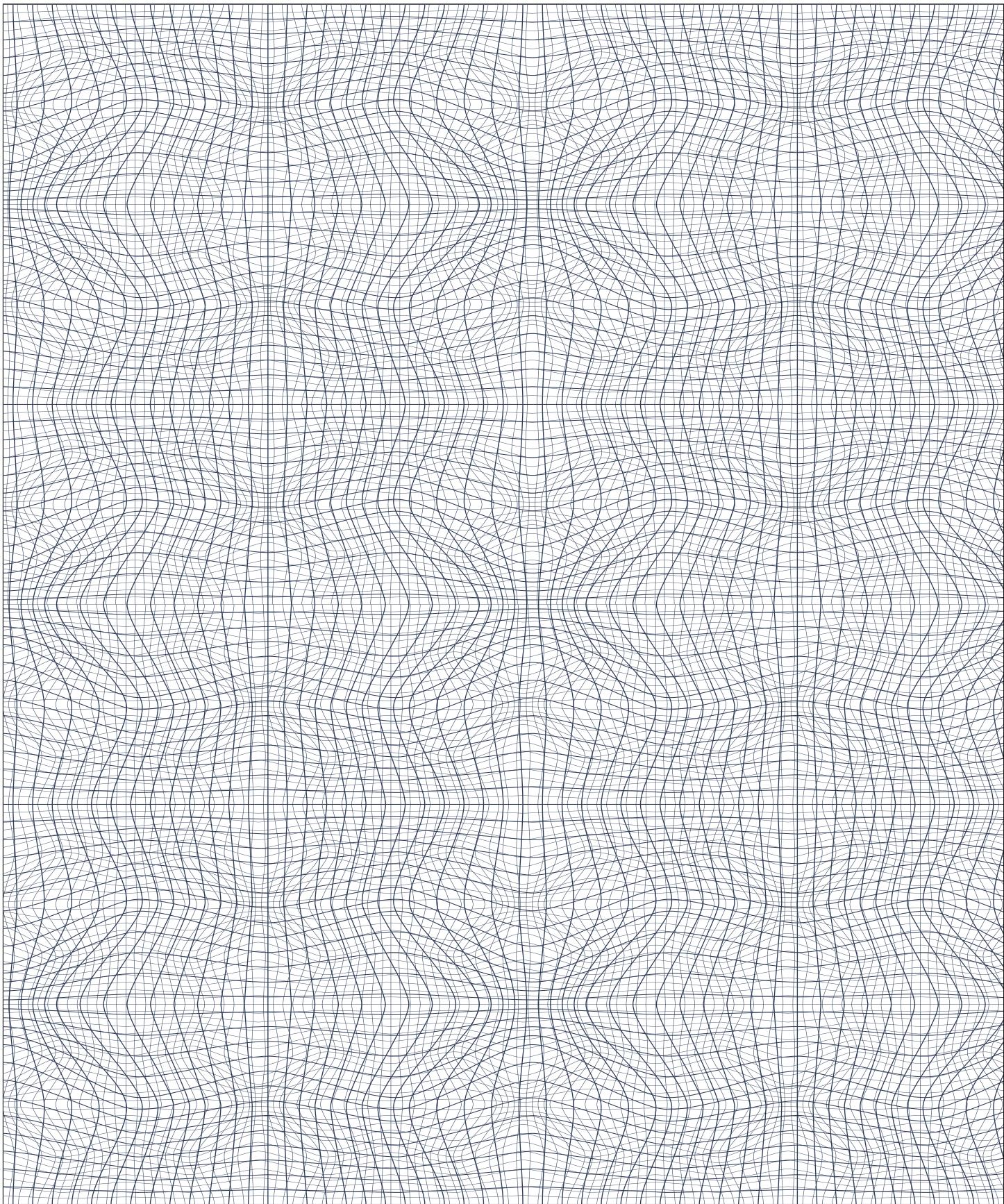
Name:

Student ID:

Instructions

1. Answer all questions.

2. Write your answers in the space provided on this paper.

3. Write your student number at the top of every answer page.

4. This is a closed-book exam: written aids are not permitted.

5. Calculators, phones and all other electronic aids are not permitted.

6. Unless otherwise specified, all code listed in this exam is correct.

7. You may assume that relevant modules such as `math` and `numpy` have been imported.

Question:	1	2	3	4	Total
Points:	10	9	8	20	47
Score:					

Programming concepts

1. Identify the programming term that best corresponds to each definition. [**10**]

   (a) What a CPU does   1

     ○ computation   ○ integer   ○ operation   ○ script   ○ slice

   (b) `while`(True)   1

     ○ call   ○ flowchart   ○ infinite loop   ○ ordinal   ○ type

   (c) A value containing a collection of characters   1

     ○ comment   ○ iteration   ○ ordinal   ○ string   ○ type

   (d) A value passed into a function call   1

     ○ algorithm   ○ function   ○ iteration   ○ literal   ○ operation

   (e) A sequence of statements in a file   1

     ○ algorithm   ○ literal   ○ modulus   ○ script   ○ slice

   (f) The + sign, for example   1

     ○ condition   ○ expression   ○ flowchart   ○ operator   ○ string

   (g) A name that is reserved by the language — you can't use it for, e.g., a variable name   1

     ○ function   ○ index   ○ iteration   ○ keyword   ○ modulus

   (h) How a computer represents a real number ($\mathbb{R}$)   1

     ○ floating-point   ○ integer   ○ memory   ○ precedence   ○ slice

   (i) Where a value can be found in a list   1

     ○ index   ○ infinite loop   ○ literal   ○ operation   ○ slice

   (j) A type with only two possible values   1

     ○ Boolean   ○ binary   ○ integer   ○ memory   ○ type

2. What will the following Python scripts print when executed? You may use the working space as you see fit, but **only write what the scripts will output in the answer box.** [9]

(a) |3|

```
s = ''

for i in range(3, 6):
    s += 'confluence'[i]

print(s)
```

Workings (optional):

Answer:

....

(b) |3|

```
s = ''

for c in 'dynamometer'[2:10:3]:
    s += c

print(s)
```

Workings (optional):

Answer:

....

(c) |3|

```
s = ''

for c in 'engulf'[::-2]:
    s += c

print(s)
```

Workings (optional):

Answer:

....

3. The following Python script is executed: [8]
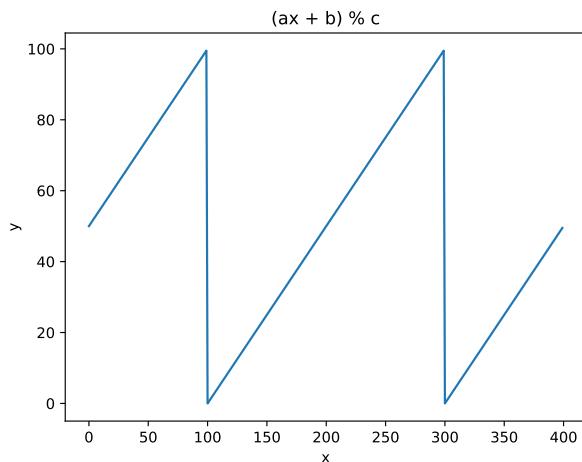
```python
x_values = array(range(n))
y_values = a * x_values + b

for x in x_values:
    y_values[x] %= c

plot(x_values, y_values)
title('(ax + b) % c')
xlabel('x')
ylabel('y')

show()
```

yielding this plot:

What must the values of a, b, c and n be?

Workings (optional):

Answer:

(a) a   _0.5_   [2]

(b) b   _50_   [2]

(c) c   _101_   [2]

(d) n   _400_   [2]

3

3. The following Python script is executed: [8]

```python
x_values = array(range(n))
y_values = a * x_values + b

for x in x_values:
    y_values[x] %= c

plot(x_values, y_values)
title('(ax + b) % c')
xlabel('x')
ylabel('y')

show()
```

yielding this plot:

What must the values of a, b, c and n be?

Workings (optional):

Answer:

(a) a   _____   [2]

(b) b   _____   [2]

(c) c   _____   [2]

(d) n   _____   [2]

3

Program synthesis

4. **Counting characters**

Write a Python script that will input a string from the user, then print the number of **unique** upper-case characters and the number of **unique** lower-case characters that appear in that input string. For example, if the user inputs the string `Hello, Holly!`, your script should print:

```
upper: 1
lower: 4
```

since there is only one unique upper-case character (H) — even though it appears twice — and there are only four unique lower-case letters(e, l, o and y). Partial credit will be given for scripts that count upper- and lower-case letters but do not take uniqueness into account, e.g., scripts that count `H` twice and `l` four times.

Workings (optional):

Answer:

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

. . . .

References

Table of character ordinal values (abridged)

Value	Character	Value	Character	Value	Character	Value	Character
0	null ('\0')	68	D	85	U	108	l
9	tab ('\t')	69	E	86	V	109	m
10	newline ('\n')	70	F	87	W	110	n
32	space (' ')	71	G	88	X	111	o
48	0	72	H	89	Y	112	p
49	1	73	I	90	Z	113	q
50	2	74	J	97	a	114	r
51	3	75	K	98	b	115	s
52	4	76	L	99	c	116	t
53	5	77	M	100	d	117	u
54	6	78	N	101	e	118	v
55	7	79	O	102	f	119	w
56	8	80	P	103	g	120	x
57	9	81	Q	104	h	121	y
65	A	82	R	105	i	122	z
66	B	83	S	106	j	128522	☺
67	C	84	T	107	k	128526	😎