Final Exam, Version 1
CSCI 127: Introduction to Computer Science
Hunter College, City University of New York

December 12, 2023

Exam Rules

- Show all your work. Your grade will be based on the work shown.

- The exam is closed book and closed notes.

- When taking the exam, you may have with you pens, pencils, and an 8 1/2" x 11" piece of paper filled with notes, programs, etc.

- You may not use a computer, calculator, tablet, smart watch, or other electronic device.

- Do not open this exam until instructed to do so.

I understand that all cases of academic dishonesty will be reported to the Dean of Students and will result in sanctions.
Name:
EmpID:
Signature:

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C		
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

(Image from wikipedia commons)

1. (a) What will the following Python code print:

   i.
   ```python
   banana = "xxyyzzBaaabbBbccc"
   print(banana.count("b"))
   ```

   Output:

   ii.
   ```python
   B = banana.split("B")
   print(B[0])
   ```

   Output:

   iii.
   ```python
   up = B[-1].upper()
   print(up)
   ```

   Output:

   iv.
   ```python
   for c in up:
       print(c.lower())
   ```

   Output:

   (b) Consider the contents of the current directory:

   ```
   banana.txt   banana.py    carrot.csv  clementine.py    dragonfruit
   ```

   i. What is the output for:

   ```
   $ ls *r*
   ```

   Output:

   ii. What is the output for:

   ```
   $   mv *.py ./dragonfruit
   $   ls
   ```

   Output:

   iii. What is the output for:

   ```
   $ ls -l | grep "banana" | wc -l
   ```

   Output:

2. (a) Select the correct option.

    i. What color is tina after this command? `tina.color("#888888")`
      ☐ black   ☐ red   ☐ white   ☐ gray   ☐ green

    ii. Select the LARGEST Binary number:
      ☐ 1011   ☐ 1101   ☐ 0111   ☐ 1010   ☐ 1110

    iii. Select the LARGEST Hexadecimal number:
      ☐ FD   ☐ EA   ☐ EF   ☐ FC   ☐ CD

    iv. What is the binary number equivalent to decimal 7?
      ☐ 1011   ☐ 0001   ☐ 1100   ☐ 0111   ☐ 1110

    v. What is the hexadecimal number equivalent to decimal 34?
      ☐ 34   ☐ 22   ☐ 24   ☐ 2B   ☐ CD

(b) Fill in the code below to make an image in which a pixel is red if it has an entry of 50 or greater in the array `elevations`. Otherwise, the pixel should be colored green.

```
# Takes elevation data of NYC and displays storm surge map
import numpy as np
import matplotlib.pyplot as plt
elevations = np.loadtxt("elevationsNYC.txt")
#Base image size on shape (dimensions) of the elevations:
mapShape = elevations.shape + (3,)
floodMap = np.zeros(mapShape)

for row in range(mapShape[0]):
    for col in range(mapShape[1]):
```
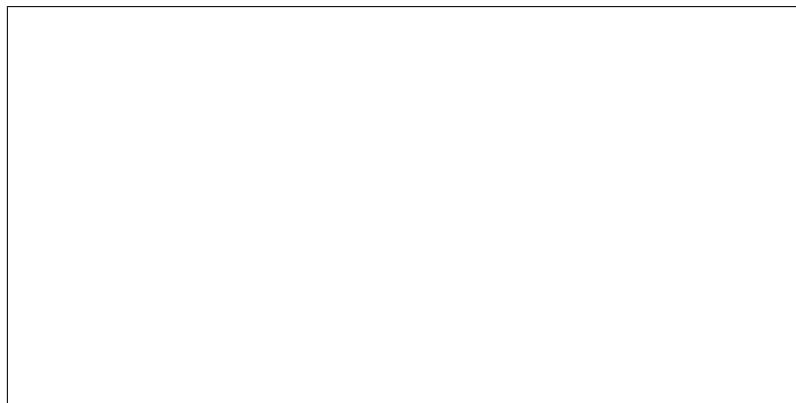
```
#Save the image:
plt.imsave("floodMap.png", floodMap)
```

3. (a) What is the value (True/False):

             `in1 = False`
  i. `in2 = False`   out =
             `out = in1 or in2`

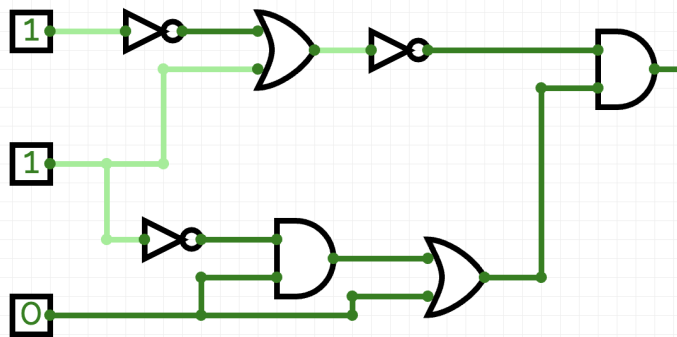   `in1 = False`
 ii. `in2 = True`   out =
             `out = not in1 or (in2 and not in1)`

             `in1 = True`
iii. `in2 = False or not in1`   out =
             `in3 = in1 and in2`
             `out = in1 or not in3`

iv.
     `in1 = True`
     `in2 = True`   out =
     `in3 = False`

(b) Design a circuit that implements the logical expression:

    `((not in1) or (in1 and in2)) and (not in3)`

4. (a) Draw the output for the function calls:
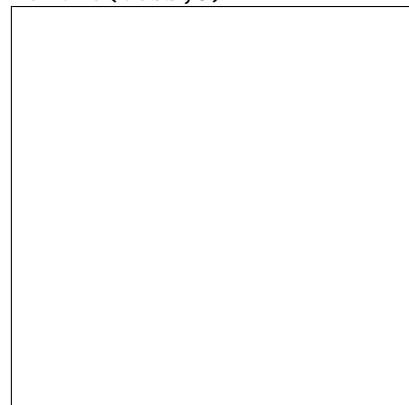
```
import turtle
tess = turtle.Turtle()
tess.shape("turtle")

def ramble(t,side):
    if side == 0:
        t.stamp()
        t.forward(50)
        t.stamp()
    else:
        for i in range(side):
            t.forward(50)
            t.left(360/side)
```

i. `ramble(tess,0)`

ii. `ramble(tess,5)`

(b) What is the output:

```
#Another mystery program...
def mystery(num):
    send = chr(num)
    if num < ord("d"):
        send = send + "H"
    return send

def enigma(letters):
    data = ""
    for x in letters:
        n = ord(x)
        c = "A"
        if n > 100:
            c = mystery(n)
        data = data + c
    return data

word = input("Enter a word: ")
s = enigma(word)
print("Output is:", s)
```

i. When the user enters: `aa`?

Output:

ii. When the user enters: `cab`?

Output:

iii. When the user enters: `alice`?

Output:

5. Fill in the Python program below. Consider the following pseudocode:

   - Ask the user for a word
   - Generate a random integer from 1 to 4 (inclusive), call it x
   - Print the user's word in reverse, with x copies of the characters on each line

Sample runs:

```
Enter a word: frog              Enter a word: frog
g                               ggg
o                               ooo
r                               rrr
f                               fff
```

```
#import the library for generating random numbers
```

```
#get user input

word =
```

```
#generate a random number from 1-4 inclusive

x =
```

```
#loop through the input word in reverse
```

```
    #prints x copies of the current character
    print(word[i]*x)
```

6. Fill in the following functions that are part of a program that analyzes star data:

- `getData()`: asks the user for the name of the CSV file and returns a DataFrame of the contents.
- `avgRadius(df)`: returns the average radius of a Hypergiant, and
- `hottestStar(df)`: returns the hottest temperature in the DataFrame.

```
import pandas as pd
def getData():
    """
    Asks the user for the name of the CSV and
    Returns a dataframe of the contents.
    """
```

```
def avgRadius(df):
    """
    Takes a DataFrame as input.
    Returns the average radius of a Hypergiant.
    First, group by "Star type" then get group "Hypergiant"
    Get the average radius of the Hypergiants by using the "Radius" column
    """
```

```
def hottestStar(df):
    """
    Takes a DataFrame as input.
    Returns the maximum value in the column, "Temperature"
    """
```

7. Fill in the Python program below that asks the user for the name of a .png (image) file and **turns the right half of the image red.** The new image should then be displayed to the user.

```
#import the libraries for images
```

[ ]

```
#get user input

infile =
```

[ ]

```
#read the image file

img =
```

[ ]

```
#get the width of the image

width =
```

[ ]

```
#make a copy of the original image

img2 =
```

[ ]

```
#set the green and blue channels to 0.0
```

[ ]

```
#set the red channel to 1.0
```

[ ]

```
#load the image into pyplot
```

[ ]

```
#display the image
```

[ ]

8. (a) Consider the following MIPS program:

    ```
    ADDI $s0, $zero, 2
    ADD $s1, $s0, $s0
    ADD $s2, $s1, $s1
    ADDI $s3, $s2, 5
    ```

    After the program runs, what is the value stored in:

    i. register `$s1`

    ii. register `$s2`

    iii. register `$s3`

    (b) What is the output for a run of this MIPS program:

    Output:

    ```
    #Loop through four letters:
    ADDI $sp, $sp, -5           # Set up stack
    ADDI $t0, $zero, 76         # Start $t0 at 76 (L)
    ADDI $s2, $zero, 80         # Use to test when you reach 80 (P)
    SETUP: SB $t0, 0($sp)       # Next letter in $t0
    ADDI $sp, $sp, 1            # Increment the stack
    ADDI $t0, $t0, 1            # Increment the letter
    BEQ $t0, $s2, DONE          # Jump to done if $t0 == 80
    J SETUP                     # If not, jump back to SETUP for loop
    DONE: ADDI $t0, $zero, 0    # Null (0) to terminate string
    SB $t0, 0($sp)              # Add null to stack
    ADDI $sp, $sp, -4           # Set up stack to print
    ADDI $v0, $zero, 4          # 4 is for print string
    ADDI $a0, $sp, 0            # Set $a0 to stack pointer for printing
    syscall                     # print to the log
    ```

9. What is the output of the following C++ programs?

(a)
```cpp
//Billy Joel
#include <iostream>
using namespace std;
int main() {
    cout << "She'll bring\nout ";
    cout << "the best and\n";
    cout << "the worst " << endl;
    cout << "you can \nbe;";
}
```

Output:

(b)
```cpp
//Mystery C++, #2
#include <iostream>
using namespace std;
int main() {
  int sum = 3;
  while (sum < 10) {
    cout << sum;
    sum = sum + sum;
  }
}
```

Output:

(c)
```cpp
//Mystery C++, #3
#include <iostream>
using namespace std;
int main() {
  for (int i = 0; i < 3; i++) {
    for (int j = 0; j <  4; j++) {
      if (j % 2 == 0) {
        cout << "+";
      } else {
        cout << "-";
      }
    }
    cout << endl;
  }
}
```

Output:

10. (a) Write a **complete C++ program** that prompts the user for a string until a non-empty string is entered. The program then prints the non-empty string that was entered.

//include library for printing and declare namespace

//main function signature

```
{
//prompt user for string until non-empty string is entered
```

```
   return 0;
}
```

(b) Write a **complete C++ program** that prints the change in population of the state of New Jersey:

$$p \quad = \quad p + (B * p) - (D * p)$$

where p is the population, B is the birth rate of 58 births for every 1000 people ($\frac{58}{1000}$) each year, and D is the death rate of 10.2 for every 1000 people ($\frac{10.2}{1000}$). In 2022, the population of New Jersey was 9.27 million. Your program should print expected population for the years 2022 to 2032. Each line should have: the year and the population (in millions).

```
//include library for printing and declare namespace
```

```
//main function signature
```

```
{
//calculate and print the predicted population
```

```
    return 0;
}
```