

Row:	SEAT:

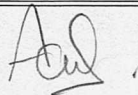
FINAL EXAM, VERSION 1  
CSci 127: Introduction to Computer Science  
Hunter College, City University of New York

20 May 2022

### Exam Rules

- Show all your work. Your grade will be based on the work shown.
- The exam is closed book and closed notes with the exception of an 8 1/2" x 11" piece of paper filled with notes, programs, etc.
- When taking the exam, you may have with you pens and pencils, and your note sheet.
- You may not use a computer, calculator, tablet, phone, earbuds, or other electronic device.
- **Do not open this exam until instructed to do so.**

*Hunter College regards acts of academic dishonesty (e.g., plagiarism, cheating on examinations, obtaining unfair advantage, and falsification of records and official documents) as serious offenses against the values of intellectual honesty. The College is committed to enforcing the CUNY Policy on Academic Integrity and will pursue cases of academic dishonesty according to the Hunter College Academic Integrity Procedures.*

I understand that all cases of academic dishonesty will be reported to the Dean of Students and will result in sanctions.									
Name: Aranda Allison									
EmpID:	2	3	9	9	3	5	4	8	
Email: allison.aranda48@myhunter.cuny.edu									
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) Fill in the code below to produce the Output on the right:

```
workdays = "1Monday2?Tuesday?Wednesday?Thursday?3"
summer_months = "*June*July*August*"
long_weekend = "Friday_Saturday_Sunday"
seasons = "+Spring+Summer+Fall+Winter"
```

i. `print( long_weekend.split("_")[-1] , workdays.split("?")[0] )`

Output:

Sunday Monday

ii. `day_list = workdays[ : ].split( " ? " )`

`print("Our week has", len( day_list ), "days.")`

Output:

Our week has 4 days.

iii. `for day in range(5)`  
`print( day_list.upper() )`

Output:

MONDAY  
TUESDAY  
WEDNESDAY  
THURSDAY

- (b) Consider the following shell commands:

```
$ ls
hello.cpp pictures pp_hello.py temp
```

i. What is the output for:  
`$ mv hello.cpp p1.cpp`  
`$ ls`

Output:

hello.cpp

ii. What is the output for:

```
$ mkdir c++ (new direct)
$ mv *.cpp c++
$ ls
```

Output:

hello.cpp pictures

iii. What is the output for:

```
$ cd c++
$ mkdir p50_60
$ mkdir pp_5
$ ls | grep pp
```

Output:

hello.cpp pictures  
pp\_hello.py temp

2. (a) Select the correct option.

i. What color is tina after this command? `tina.color(0.0,0.0,0.0)`

- ☒ black    ☐ red    ☐ white    ☐ gray    ☐ purple

ii. Select the LARGEST Binary number:

- ☐ 0110    ☐ 1001    ☒ 1101    ☐ 1011    ☐ 0000

iii. Select the SMALLEST Hexadecimal number:

- ☐ 0A    ☐ 22    ☐ A0    ☒ FF    ☐ CD

iv. What is the Binary number equivalent to decimal 22?

- ☐ 11010    ☐ 01110    ☒ 10110    ☐ 00011    ☐ 10101

v. What is the Hexadecimal number equivalent to decimal 20?

- ☒ 14    ☐ A1    ☐ F0    ☐ 1F    ☐ 18

(b) Fill in the code to produce the Output on the right:

`nums = [ 23, 45, 76, 23, 98, 45, 11, 4, 33, 29, 5, 66 ]`

i. `for i in range(  ,  ):`  
`print(nums[i], end=" ")`

Output:

76 23 98

ii. `for j in range(  ,  ,  ):`  
`print(nums[j], end=" ")`

Output:

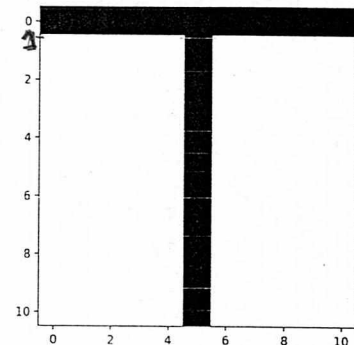
45 23 45 4

`import numpy as np`  
`import matplotlib.pyplot as plt`  
`img = np.ones( (11,11,3) )`

iii. `img[  ,  , :] = 0 # black row`

`img[  ,  , :] = 0 # black column`  
`plt.imshow(img)`  
`plt.show()`

Output:



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

in1 = False

i. in2 = True

out = not (in1 and in2) and (not in1 and in2)

☒ True

☐ False

in1 = False

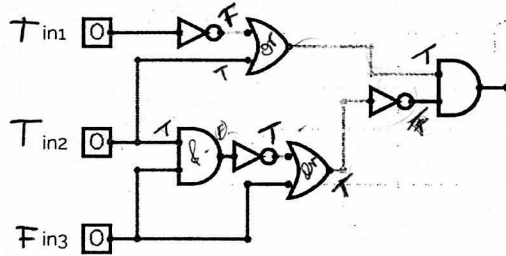
in2 = True

ii. in3 = not (not in1 or not in2)

out = (not in1 or not in2) and not (in2 or in3)

☐ True

☒ False



iii.

in1 = True

in2 = True

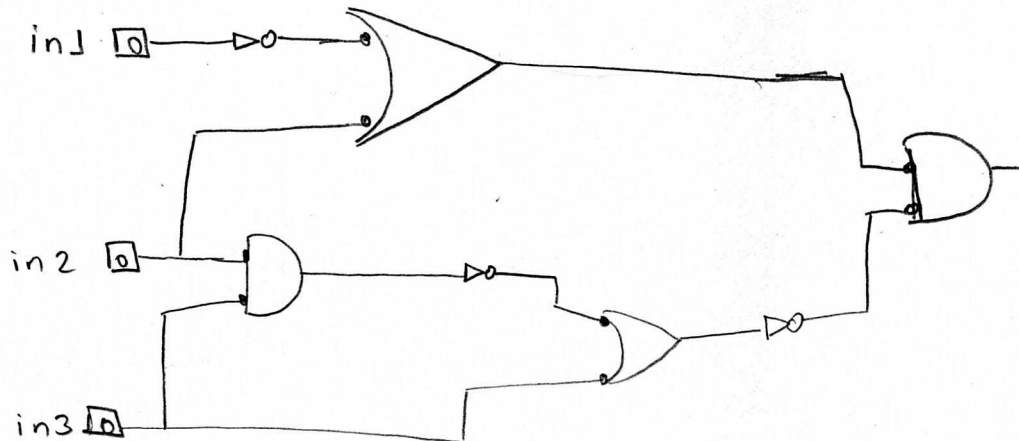
in3 = False

☐ True

☒ False

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

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





4. Consider the following functions:

```
def jig(n, m):  
    for i in range(n):  
        if i % 3 == 0:  
            print(saw(i, m))
```

```
def saw(i, m):  
    for j in range(i):  
        m+=1  
    return m
```

```
def main():  
    jig(10, 5)
```

10	10
9	10
8	10
7	8
6	6

(a) What are the formal parameters for saw()?

i, m

(b) What are the actual parameters for jig?

10, 5

(c) How many calls are made to saw() after calling main()?

9

(d) What is the output after calling main()?

Output:

10 8 6

5. A palindrome is a string (word or sentence, e.g. "madam" or "nurses run") that reads the same backward as forward. Design an algorithm that reads a string and outputs whether it is a palindrome or not. You must write detailed pseudocode as a precise list of steps that completely and precisely describe the algorithm.

Libraries

(if  
any):

no necessary

Input:

input a string

Output:

a palindrome string or not.

Principal Mechanisms (select all that apply):

- ☒ Single Loop      ☐ Nested Loop      ☒ Conditional (if/else) statement  
☐ Indexing / Slicing      ☐ split()      ☒ input()

Process (as a concise and precise LIST OF STEPS / pseudocode):

(Assume libraries, if any, have already been imported.)

→ ASK the user to input a string  
→ name the input; call it S  
→ use a single loop to read every letter in S.  
if the string is a palindrome or  
not, print it anyway.

6. Consider the `art_library.csv` from the Art Garfunkel's Library from kaggle. Each row in the dataset corresponds to a book. A snapshot of the data is given in the image below:

Date Read	Author	Books	Year Published	Pages	Favorite
Jun-68	Jean-Jacques Rousseau	The Confessions	1781	606	1
Jun-68	Erich Fromm	The Art of Loving	1956	146	1
Jun-68	Mark Twain	The Adventures of Huckleberry Finn	1884	288	0
Jul-68	James Thurber	My Life and Hard Times	1933	115	0
■ ■ ■					
Jan-22	James Michener	Caravans	1963	320	0
Jan-22	Abraham Lincoln	The Spiritual Growth of a Public Man	1973	47	0
Feb-22	Joe Scarborough	Saving Freedom	2020	272	0

Fill in the Python program below:

#Import the libraries for data frames

```
import pandas as pd
```

#Prompt user for input file name:

```
csvFile = input("art_library.csv")
```

#Read input data into data frame:

```
lib = pd.read_csv(csvFile)
```

#Print the number of applications for date

```
print(lib["Date Read"].value_counts())
```

#Group the data by author to extract books written by Jean-Jacques Rousseau  
#use groupby and get\_group

```
rousseau = lib.groupby("Jean-Jacques Rousseau")
```

#Print the latest year a Rousseau book was published

```
print(rousseau["Year Published"].max())
```



7. Fill in the following functions that are part of a program that extracts data from a CSV file:

- `getData()`: asks the user for the name of the CSV and returns a DataFrame of the contents.
- `extract()`: computes and returns the maximum, minimum and average value of the input column
- `getList()`: returns a list of length  $(\text{max}-\text{min})/\text{avg}$ , containing equally spaced numbers in range  $[\text{min}, \text{max}]$

```
import pandas as pd
```

```
def getData():
```

```
    """
```

```
    Asks the user for the name of the CSV and
```

```
    Returns a dataframe of the contents.
```

```
    """
```

```
name = input("Enter CSV file:")
df = pd.read_csv(name)
return df
```

```
def extract(df, col):
```

```
    """
```

```
    Computes and returns the maximum, minimum and average values
```

```
    of the column col in dataframe df
```

```
    """
```

```
max = max.value
min = min.value
mean = mean.value
return (max, min, mean)
```

```
def getList(max, min, avg):
```

```
    """
```

```
    Creates and returns a list of equally spaced numbers in range [min, max].
```

```
    The length of the list is  $(\text{max}-\text{min})/\text{avg}$ 
```

```
    """
```

```
for list in range [min, max]:
    length = (max - min) / avg
return (getList)
```

8. (a) What is printed by the MIPS program below:

Output:

AAAAAAAAAAAA

- (b) Modify the program to print out "ACEGIK". Shade in the box for each line that needs to be changed and rewrite the instruction below, or add instructions where necessary.

- ☒ ADDI \$sp, \$sp, -15<sup>7</sup> # Set up stack
- ☐ ADDI \$s3, \$zero, 1 # Store 1 in a register
- ☐ ADDI \$t0, \$zero, 65 # Set \$t0 at 65 (A)
- ☐ ADDI \$s2, \$zero, 15 # Use to test when you reach 15
- ☐ SETUP: SB \$t0, 0(\$sp) # Next letter in \$t0
- ☐ ADDI \$sp, \$sp, 1 # Increment the stack
- ☐ ADDI \$s3, \$s3, 1 # Increment the counter by 1
- ☐ BEQ \$s3, \$s2, DONE # Jump to done if \$s3 == 14
- ☐ 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, -14<sup>6</sup> # 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. Fill in the C++ programs below to produce the Output on the right.

```

#include <iostream>
using namespace std;
int main()
{
    for( i=2 ; i <=15; i+=2 ){
        cout << i+2 << endl;
    }
    return 0;
}

```

(a)

Output:

4 6 } 2  
6 8 } 2  
8 10 } 2  
10 12 } 2  
12 14 } 2  
14 16 } 2

```

#include <iostream>
using namespace std;
int main()
{
    int n=-4, m=10;
    while( n+m >=0 ){
        n--;
        m+=2;
        cout << n << " " << m << endl;
    }
    return 0;
}

```

(b)

Output:

n m  
-5 12  
-6 14  
-7 16  
-8 18

```

#include <iostream>
using namespace std;
int main(){
    for ( i=515 ; i>=510 ; i-- ){
        for ( j=i+101 ; j=i+102 ; j-- ){
            cout << i << i+j << " ";
        }
        cout << endl;
    }
    return 0;
}

```

(c)

Output:

515 514 513 512 511 510  
616 615 614 613 612  
717 716 715 714  
818 817 816  
919 918

10. (a) Write a complete C++ program that repeatedly asks the user for their age until the age is in range [18, 65], then it outputs the age:

```
//include library and namespace
```

```
#include <iostream>
using namespace std;
```

```
//main function signature
```

```
int main ()
```

```
{
```

```
    //variable initialization
```

```
    int age;
```

```
    //repeatedly ask for age until in [18, 65]
```

```
    cout << "enter age:";
    cin >> age;
    if (age <= 18)
        cout << "enter age again: \n";
    else if (age >= 65)
        cout << "enter age again: \n";
    else
        cout << "age \n";
```

```
    //output age
```

```
    cout << age << endl;
```

```
    return 0;
```

```
}
```

- (b) The global population has grown from 1 billion in 1800 at a rate of approximately 1.1% per year.

Write a complete C++ program that asks the user for a year after 1800 and returns the global population (in billions) in that year.

```
//include library and namespace
```

```
#include <iostream>
using namespace std;
```

```
//main function signature
```

```
int main()
```

```
{
```

```
//declare variables
```

```
int p = 1800.
int year
```

```
//obtain input
```

```
cout << "Enter a year after 1800: \n";
cin >> year
```

```
//compute the population at 1.1% yearly increase
```

```
p = year * 0.011;
```

```
//Output the global population (in billions) during the year entered by the user
```

```
cout << "p * 1000000 \n";
      (billions)
```

```
return 0;
```

```
}
```

SCRATCH PAPER