

CS 101 Practice Final Exam

- **Be sure to enter your NetID and the code below on your Scantron.**
- Do not turn this page until instructed to.
- This is a 120-minute exam.
- There are 32 questions:
 - 12 MATLAB multiple-choice questions worth 5 points each;
 - 18 Python multiple-choice questions worth 5 points each; and
 - 2 coding questions worth 25 points eachfor a total of 200 possible points.
-
- Each multiple choice question has only *one* correct answer.
- You must not communicate with other students during the exam.
- No books, notes, or electronic devices are permitted during the exam.

1. Fill in your information:

Full Name: _____

UIN (Student Number): _____

NetID: _____

Lab Section: _____

2. Fill in the following answers on the Scantron form:

Zone 1

The following 12 questions involve MATLAB.

1/1. (5 points) Consider the following MATLAB program:

```
x = [ 1 2 ];  
y = [ 3 4 ];  
z = [ x' [ y ; y ] ]';
```

What is the **value** of **z** after this program executes?

A. $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 3 & 4 \end{bmatrix}$

B. $\begin{bmatrix} 1 & 3 & 3 \\ 2 & 4 & 4 \end{bmatrix}$

C. $\begin{bmatrix} 1 & 3 & 4 \\ 2 & 3 & 4 \end{bmatrix}$

D. ★ $\begin{bmatrix} 1 & 2 \\ 3 & 3 \\ 4 & 4 \end{bmatrix}$

E. None of the other answers are correct

Solution.

2/1. (5 points) Consider the following MATLAB program:

```
A = ones( 4,4 ) + 2 * eye( 4,4 );
```

```
A = A - 2;
```

```
A( :,3:4 ) = A( :,3:4 ) + 2;
```

What is the **value** of **A** after this program executes?

A. ★ $\begin{bmatrix} 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & -1 & 3 & 1 \\ -1 & -1 & 1 & 3 \end{bmatrix}$

B. $\begin{bmatrix} 0 & -2 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ -2 & -2 & 2 & 0 \\ -2 & -2 & 0 & 2 \end{bmatrix}$

C. $\begin{bmatrix} -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -3 & 1 \\ -1 & -1 & 1 & -3 \end{bmatrix}$

D. $\begin{bmatrix} 1 & 1 & 3 & 3 \\ 1 & 1 & 3 & 3 \\ 1 & 1 & 3 & 3 \\ 1 & 1 & 3 & 3 \end{bmatrix}$

E. None of the other answers are correct

Solution.

3/1. (5 points) Consider the following MATLAB function stored in `prink.m`:

```
function [ f g ] = prink( x,y )
    f = x .^ -0.5;
    g = y + f .* 2;
end
```

Which of the following correctly assigns the results of a call to `prink f` to `F` and `g` to `G`, respectively?

- A. `F,G = prink(2,3);`
- B. `[F G] = prink([2 3]);`
- C. `[F G] = prink(2 3);`
- D. `[F G] = prink [2 3];`
- E. ★ `[F G] = prink(2,3);`

Solution.

4/1. (5 points) Recollect that MATLAB represents polynomials as an array of coefficients from the highest-order coefficient to the lowest. For instance,

$$x^3 + x + 2$$

is written as the array [1 0 1 2].

How would we represent the summation of the two polynomials

$$x^2 + x + 1$$

and

$$x^3 + x^2 - 1$$

as a MATLAB polynomial array?

- A. [1 1 1] + [1 1 0 -1]
- B. [1 1 1] + [-1 0 1 1]
- C. ★ [0 1 1 1] + [1 1 0 -1]
- D. [1 1 1 0] + [1 1 0 -1]

Solution.

5/1. (5 points) Consider the following two-dimensional MATLAB array, stored in the variable **A**:

$$\begin{bmatrix} 1 & 1 \\ 3 & 6 \\ 9 & 36 \\ 27 & 216 \end{bmatrix}$$

How can we index and retrieve the value 36 from this array?

- A. `A(2,1)`
- B. ★ `A(3,2)`
- C. `A(2,3)`
- D. `A[2,1]`
- E. `A[3,2]`

Solution.

6/1. (5 points) For this problem, you should compose a function which accomplishes a given task using the available code blocks arranged in the correct functional order.

The Levi-Civita symbol represents a direction of travel through the permutations of the elements of a vector (or higher-order tensor). For a 3D vector, given numbers i, j, k ,

$$\varepsilon_{ijk} = \begin{cases} +1 & (1, 2, 3), (2, 3, 1), (3, 1, 2); \\ -1 & (3, 2, 1), (2, 1, 3), (1, 3, 2); \\ 0 & \text{if } i = j, j = k, \text{ or } k = i. \end{cases}$$

Compose a function `epsilon` which accepts a vector `ind` representing the three values i, j, k in the above definition. The function should return the value of ε_{ijk} for those values.

```

1 end
2 s = 0;
3 s = -1;
4 s = +1;
5 if ind(1) == ind(2) | ind(2) == ind(3) | ind(3) == ind(1)
6 elseif ind(1) == 1 & ind(2) == 2 & ind(3) == 3
7 elseif ind(1) == 2 & ind(2) == 3 & ind(3) == 1
8 elseif ind(1) == 3 & ind(2) == 1 & ind(3) == 2
9 elseif ind(1) < ind(2) & ind(2) < ind(3) & ind(3) < ind(1)
10 function [ s ] = epsilon( ind )
11 function epsilon( ind )
12 else
13 if ind(1) == 1 & ind(2) == 2 & ind(3) == 3

```

- A. 10, 5, 2, 6, 3, 7, 3, 8, 3, 12, 4, 1
- B. ★ 10, 5, 2, 6, 4, 7, 4, 8, 4, 12, 3, 1, 1
- C. 10, 2, 13, 4, 12, 3, 1, 1
- D. 11, 5, 2, 6, 4, 7, 4, 8, 4, 12, 3, 1

Solution.

7/1. (5 points) Consider the following MATLAB program:

```
s = (2 < 3) & ((2 > 3) | (1 ~= 0))
```

What is the final value of `s`?

A. True

B. ★ 1

C. 0

D. false

Solution.

8/1. (5 points)

```
x = eye( 2,2 );  
y = [ x(:,1) x(:,2) ];  
A = [ x y ; y x ];
```

What is the final value of `A(1:2,1:2)`?

- A. [0 1 ; 1 0]
- B. [1 1 ; 1 1]
- C. [0 0 ; 0 0]
- D. ★ [1 0 ; 0 1]

Solution.

9/1. (5 points)

```
x = linspace( -1,1,101 );  
y1 = exp( x );  
y2 = tan( x );  
y3 = rand( 1,numel(x) );
```

How would you successfully plot all three of these data series as points? (Assume any given plot format strings are valid.)

A. ★

```
plot( x, y1, 'r.', x,y2, 'g.', x,y3, 'b.' );
```

B. `plot(x, y1, 'r.', y2, 'g.', y3, 'b.');`

C. hold on;

```
plot( x, y1 );  
plot( x, y2 );  
plot( x, y3 );
```

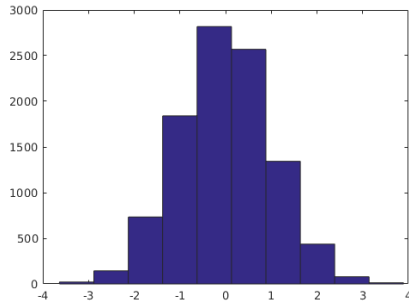
D. `plot(x,y1, x,y2, x,y3);`

Solution.

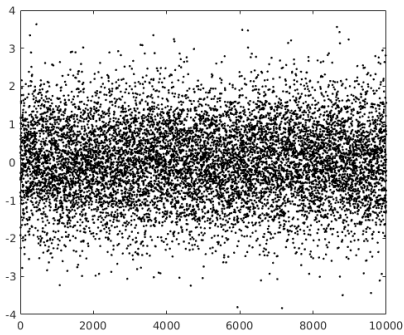
10/1. (5 points) Consider the following program, which produces 10,000 random numbers selected from a certain distribution and plots them:

```
x = randn( 10000,1 );  
plot( x,'k.' );
```

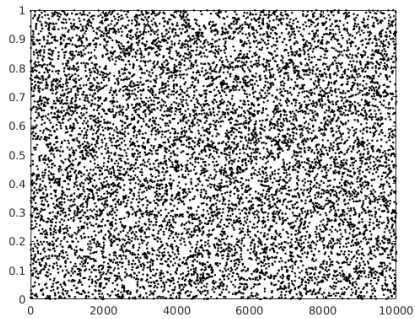
Which of the following plots could result from executing this program?



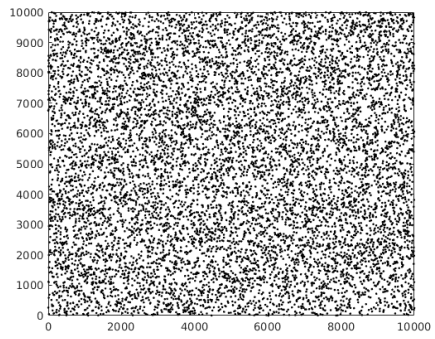
A.



B. ★



C.



D.

Solution.

11/1. (5 points)

```
A = eye( 4,4 );  
for x = 1:2:4  
    A( x,x ) = 0;  
end
```

What is the final value of A?

A. $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

B. ★ $\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

C. $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

D. $\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$

E. $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Solution.

12/1. (5 points)

```
A = eye( 3,3 ) - ones( 3,3 );
for x = 1:3
    for y = 1:3
        if x <= y
            A( x,y ) = mod( x,y );
        end
    end
end
```

What is the final value of A?

A. $\begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$

B. ★ $\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 2 \\ -1 & -1 & 0 \end{bmatrix}$

C. $\begin{bmatrix} 2 & -1 & -1 \\ 3 & 2 & -1 \\ 4 & 5 & 2 \end{bmatrix}$

D. $\begin{bmatrix} 1 & 1 & 0 \\ 2 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$

E. $\begin{bmatrix} -1 & 1 & 1 \\ -1 & -1 & 2 \\ -1 & -1 & -1 \end{bmatrix}$

Solution.

Zone 2

The following 18 questions involve Python.

13/1. (5 points) Consider the following incomplete Python program:

```
a = 'BUSTER'  
b = 'KEATON'  
d = { }  
for x,y in zip( a,b ):  
    ???  
s = ''  
for c in a:  
    s += d[ c ]
```

What should replace the three question marks to cause this program to yield a final value for `s` of 'BUSTER'?

- A. ★ `d[x] = y`
- B. `d[y] = x`
- C. `d[a] = b`
- D. `d[b] = a`
- E. `d[a] = x`

Solution.

14/1. (5 points) Consider the following Python program:

```
d = { 0:0,1:0,2:0,3:0 }  
for i in range( -3,7 ):  
    d[ i%4 ] += i  
x = d[ 0 ]
```

What is the final *value* of *x*?

- A. 2
- B. 3
- C. ★ 4
- D. 5
- E. 6

Solution.

15/1. (5 points) Consider the following Python program:

```
d = { "M":1, "A":1, "R":2, "X":1 }  
for c in "HARPO":  
    print( d[ c ] + '-' )
```

What kind of exception will this program throw?

- A. ★ `KeyError: 'H'`
- B. `TypeError: list indices must be integers, not str`
- C. `SyntaxError: invalid syntax`
- D. `TypeError: unsupported operand type(s) for +: 'int' and 'str'`

Solution.

16/1. (5 points) Consider the following Python program:

```
e = list( range( 0,10,2 ) )
d = [ '0','0','0','0' ]
for i in range( 0,len(e) ):
    d[ i%4 ] += str( e[ i ] )
x = d[ 1 ]
```

What is the final *type* of x?

- A. ★ str
- B. list
- C. int
- D. dict

Solution.

17/1. (5 points) Consider the following incomplete Python program:

```
sum = 0
???:
    sum += i + 1
```

The program is intended to sum all of the integers between 1 and 10 (inclusive). What should replace the three question marks to complete the program?

- A. ★ `for i in range(0,10)`
- B. `while i <= 10`
- C. `for i in range(1, 11)`
- D. `while i in range(10)`

Solution.

18/1. (5 points)

```
x = np.array( [ 4,2 ] * 2 )
```

What is the final *value* of `x`?

A. $\begin{bmatrix} 4 & 4 \\ 2 & 2 \end{bmatrix}$

B. $\begin{bmatrix} 4 \\ 2 \\ 4 \\ 2 \end{bmatrix}$

C. ★ $[4 \ 2 \ 4 \ 2]$

D. $[8 \ 4]$

Solution.

19/1. (5 points)

```
import itertools
x = 'hardy'
???
    print( x )
```

Replacing the three question marks with which of the following will result in 'hardy' being printed exactly ten times?

- A. `for a in itertools.combinations(x,5):`
- B. `for a in itertools.combinations(x,2):`
- C. ★ `for a in itertools.combinations(x,3):`
- D. `for a in itertools.combinations(x,4):`

Solution.

20/1. (5 points) Consider the following incomplete Python program:

```
y = 1.0 # initial position, m
v = 0.0 # initial velocity, m/s
g = -9.8 # acceleration due to gravity, m/s^2
t = 5.0 # initial time, s
dt = ??? # time increment, s

while y > 0.0:
    t += dt
    v += g * dt
    y += v * dt
```

Which of the following values for `dt` will yield the most accurate solution?

- A. ★ `dt = 1e4`
- B. `dt = 0.1`
- C. `dt = 1e-4`
- D. `dt = 10`

Solution.

21/1. (5 points)

```
s = 'STOUGE'  
x = ''  
for i in range( 0,len( s ) ):  
    if ( i>1 ) and ( i<4 ):  
        x = s[ i:i+2 ] + x
```

What is the *value* of `x` after this program is executed?

- A. ★ 'OG00'
- B. '00T0'
- C. '00'
- D. 'T0'
- E. None of the other answers are correct.

Solution.

22/1. (5 points)

```
def prod_pairs( A ):
    total = 0
    ???
    return total
```

The function `prod_pairs` accepts a list of floats named `A`. `prod_pairs` should return the product of each pair of values in `A` (without repeats). For example, given the list `[1,2,3]`, `prod_pairs` should return 11 from $(1*2)+(1*3)+(2*3) = 11$. What should replace the three question marks to complete the function? (Assume any necessary `imports` to have taken place already.)

A. ★

```
for i in range( len( A ) ):
    for j in range( i+1,len( A ) ):
        total += A[ i ] * A[ j ]
```

B. for i in range(len(A)):
for j in range(len(A)):
total *= A[i] * A[j]

C. for i,j in enumerate(A):
total += A[i] * A[j]

D. for i in itertools.permutations(A):
total += i[0] * i[1]

Solution.

23/1. (5 points) What do we call the optimization heuristic that involves choosing the best from a stochastically sampled subset of the domain?

- A. Brute-force search
- B. Local optimum
- C. Hill climbing
- D. ★ Random search

Solution.

24/1. (5 points)

```
def most_sessions( datafile ):
    d = { }
    for line in open( datafile ):
        ???
    return d
```

The function `most_sessions` should compute the greatest number of events per session of each user in the database by reading a comma-separated value input file of session data. The result should be returned from the function as a dictionary. The first column of each line in the input file is expected to contain the user's name represented as a string. The second column is expected to contain an integer representing the events in a session. Here is an example input file:

```
Groucho,10
Beppo,12
Harpo,25
Your program should ignore a non-conforming line like this one.
Beppo,3
Chico,6
Chico,14
```

The resulting return value for this file should be the following dictionary:

```
{ 'Beppo':12, 'Harpo':25, 'Groucho':10, 'Chico':14 }
```

What should replace the three question marks to complete the function?

A. ★

```
try:
    s,f = line.split( "," )
    if s not in d:
        d[ s ] = 0
    if int( f ) > d[ s ]:
        d[ s ] = int( f )
except:
    continue
```

B. if line not in d:
d[line] = 0

```
try:
    s,f = line.split( "," )
except:
    d[ s ] = int( f )
    continue
```

C. try:
s,f = line.split(",")
except:
continue
if f not in d:
d[f] = 0
d[f] = max(int(s), d[f])

```
D.  try:
      s,f = line.split()
      d[ s ] = int( f )
    except:
      break
```

Solution.

25/1. (5 points)

```
s = ''.join( [ "3","3","2","2" ] )  
x = 0  
for i in range( len( s )-1 ):  
    x += int( ??? )
```

What should replace the three question marks so the resulting value of x is 7?

- A. `s[i:i+2:i]`
- B. ★ `s[i+1:i:-1]`
- C. `s[i+3:i:-1]`
- D. `s[i+1:i+2]`

Solution.

26/1. (5 points)

```
x = [ ]
for i in range( 1,11 ):
    for j in range( i+1,11 ):
        t = i,j
        x.append( t )
```

After the program runs, which of the following is an element of `x`?

- A. ★ (9,10)
- B. (9,3)
- C. (4,1)
- D. (8,7)
- E. (10,4)

Solution.

27/1. (5 points)

```
e = [ 5,4,3,2,1,10,9,8,7,6 ]  
d = { 0:0,1:0,2:0,3:0 }  
for a,b in enumerate( e ):  
    d[ a%4 ] += b  
x = d[ 1 ]
```

After it is run, what is the final *value* of **x**?

- A. 4
- B. ★ 20
- C. 14
- D. 26
- E. 8

Solution.

28/1. (5 points)

```
x = "2,2,a".split(',')
x.reverse()
try:
    print( x[-1] )
except:
    print( type( len( x ) ) )
```

After it is run, what is printed by this program?

- A. TypeError
- B. 3
- C. ★ 2
- D. NoneType (the type of None)

Solution.

Zone 3

29/1. (25 points)

Your friend Vanessa can't remember her Facebook password and wants your help figuring it out. She remembers the password is exactly 8 characters long. She also remembers that her username is either "vanessa" or "VanessaC" or "Vanessa95". Assume someone else has already written a function `login` that takes a two string arguments representing a username and password combination. `login` returns `True` if the input username and password are valid credentials for Facebook and `False` otherwise. Your function `guess_password` should perform a brute force search and return the correct username and password for Vanessa's account as a tuple of two strings.

We set up the alphabet string for you. Assume all of the possible password characters are contained in this string. You may `import itertools` in your solution if you prefer, but no other libraries are allowed.

```
def guess_password():
    alphabet="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
    alphabet+="0123456789!@#$%^&*()-_+=,<.>/?~`"
```

Solution.

Zone 4

30/1. (25 points) Write a Python program to simulate a population of rabbits for 50 years. Your simulation should update annually (i.e. $\Delta t = 1$ year). The initial population of rabbits is 15. Each year, 5% of the population of the previous year dies off, and each year exactly 5 new rabbits are born. In your simulation, it should be impossible for “partial rabbits” to exist. For example, the population should never be 13.7—we’re not interested in 70% of a rabbit. You should always round the population *down* to a whole number.

You may `import numpy` in your solution if you prefer, but no other libraries are allowed.

Solution.

Zone 5

31/1. (25 points) Consider the Taylor series definition of the exponential function:

$$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \frac{x^7}{7!} + \dots$$

The series converges for all real x , so to calculate $\exp(x)$ to within a few decimal places of accuracy one just needs to include sufficient terms in the calculation.

The following MATLAB function `expp` was written in order to calculate the value of $\exp(x)$ for all x to three decimal places of accuracy (`atol` in the code). Translate this function into a Python function—also called `expp`—which yields identical output from the function as the MATLAB function for given input. You may `import numpy as np` in your solution if you prefer, but no other libraries are allowed. (Assume a valid NumPy-compatible function `factorial` is also available. Also recall that $0! = 1! = 1$.)

```
function [ y ] = expp( x )
    y = 0;
    yold = 1;
    n = 0;
    atol = 1e-3; % tolerance
    while ( abs( y-yold ) > atol )
        yold = y;
        term = ( x .^ n ) / factorial( n );
        if (mod(n,2) == 1)
            term = -term;
        end
        y = y + term;
        n = n + 1;
    end
end
```

Solution.
