

Mid-term Exam

● Graded

Student

Krishna Jaiswal

Total Points

49.5 / 89 pts

Question 1

Browser error analysis

0 / 4 pts

- + 1 pt Browser cannot verify site's certificate because the issuing Certificate Authority (CA) is unknown/untrusted. (failed chain, MITM risk)
- + 1 pt safe "Go Back": mention steps like (go back, verify URL, network, report the website or any valid measure that could be taken)
- + 2 pts cautious "Continue" condition (verifying or trust the issuer/website externally / internal host / no sensitive actions)

✓ + 0 pts Incorrect/ Unattempted

- 1 have to mention untrusted or unknown CA; measures after go back ??
- 2 mention measures like check url,etc
- 3 havent mentioned under what conditions (like dont enter confidential info) and steps ?!

Question 2

Values after statement execution

4.5 / 5 pts

✓ + 0.5 pts If 10 out of 11 are are correct, full marks are assigned. Similarly, minimum is 0. Hence, turn -.5 to 0.

- 0.5 pts value of row 1 x incorrect/incomplete
- 0.5 pts value of row 2 x incorrect/incomplete
- 0.5 pts value of row 2 y incorrect/incomplete
- 0.5 pts value of row 3 lxy incorrect/incomplete
- 0.5 pts value of row 3 txy incorrect/incomplete
- 0.5 pts value of row 4 lxy2 incorrect/incomplete
- 0.5 pts value of row 4 txy2 incorrect/incomplete
- 0.5 pts value of row 5 lxy incorrect/incomplete

✓ - 0.5 pts value of row 5 lxy2 incorrect/incomplete

✓ - 0.5 pts value of row 6 txy incorrect/incomplete

- 0.5 pts value of row 6 txy2 incorrect/incomplete
- 0 pts All correct

Question 3

Value of expressions

3 / 4 pts

- 1 pt If (a) is incorrect/unattempted.

Correct Answer: (a) False , True==False will return False as they are not same.

✓ - 1 pt If (b) is incorrect/unattempted.

Correct Answer: (b) SyntaxError , Explanation- It is an invalid expression

- 1 pt If (c) is incorrect/unattempted.

Correct Answer: (c) True, not True==False is interpreted as "not (True==False)" then "not False" so it will return True

- 1 pt If (d) is incorrect/unattempted.

Correct Answer: (d) True, 10==5+5>False is interpreted as 10==10>False then True>False which will return True

- 4 pts All Incorrect/Unattempted

- 0 pts All Correct

Question 4

Values after statement execution in given program

3 / 5 pts

LDIG: 1

+ 0 pts Click here to replace this description.

✓ + 0.5 pts t1: (0, 1, -d, 1, -d)

✓ + 0.5 pts t2:(0, 1, -d)

+ 0.5 pts l1: [(0, 1,-d),d]

+ 0.5 pts l2: [0,(0,1,-d),d]

+ 0.5 pts l1:[(0,1,-d),d,[0,(0,1,-d),d]]

+ 0.5 pts l2:[0,(0,1,-d),d]

✓ + 0.5 pts l2: [0,1,d*d]

✓ + 0.5 pts t2: (0,1,-d)

✓ + 0.5 pts l2: [1,d*d]

✓ + 0.5 pts l2: [1,-d*d]

+ 0 pts All Incorrect/ Not attempted

💬 Replace "d" with your LGID to get the answer for each list and tuple.

Question 5

Program output

0 / 3 pts

+ 1 pt a) Cause of error mentioned (details given):
Or
If full formal name of error is given:

+ 1 pt b) [LDIG,2,3]

+ 1 pt c) ALPHA

✓ + 0 pts None of them is correct

Question 6

Error finding in program

3 / 5 pts

✓ + 1 pt None

+ 1 pt 2 nd question
syntax error or if error description is correct

✓ + 1 pt 3rd question
name error or if error description is correct

✓ + 0.5 pts 4th question
Index error or if error description deals with indexing

✓ + 0.5 pts 4 th question,
identifying Error: This statement does nothing. or It does not assign a value, print anything, or modify
any variable, out of index

+ 1 pt 5th question
TypeError or if error description is correct

+ 0 pts if everything is wrong

Question 7

Result of program execution

2 / 4 pts

```
input1 = input('input1: ')
print('user' + input1)
```

✓ + 1 pt user1,2\$

+ 0.5 pts Missing required symbols (@, \$)

```
input1 = input('input: ')
print('user', input1)
```

✓ + 1 pt user@1,2\$

+ 0.5 pts Missing required symbols (@, \$)

```
input1 = input('input: ')
print('user', input1.split())
```

+ 1 pt user@['1,2']\$

+ 0.5 pts Missing required symbols (@, \$, [], ' ')

```
input1 = input('input: ')
print('user' + input1.split())
```

+ 1 pt TypeError or error description correct

+ 0 pts All incorrect answers

Question 8

Python code to compute grade

4.5 / 6 pts

- 0 pts Syntax error. Relational operators like ">=" and "<=" cannot be written like that.

- 0 pts Syntax error. Expressions like "a/b" or "3/4" cannot be written like that.

✓ - 0.5 pts Adjusting for floating point precision error in calculation of 3/4th of total_days: missing or not done correctly.

- 1 pt 'if-else' condition not written correctly.

Question says "...less than or equal to..." but ">=" has been written in 'if' condition instead of ">".

- 1 pt 'if-else' condition not written correctly.

Question says "...less than or equal to..." but "<" has been written in 'if' condition instead of "<=".

- 1 pt 'if-else' condition not written correctly.

Question says "...less than or equal to..." but ">" has been written in 'if' condition instead of ">=".

✓ - 1 pt Logical error in if-else branch.

- 3 pts if-else branch incomplete, completely missing or incorrect.

- 1 pt Logical error in if-else branch.

- 1 pt Logical error in if-else branch.

- 1.5 pts Lowering the grade by 1: missing or not done correctly.

- 1 pt Checking that the lowered grade is not negative: missing or not done correctly.

- 6 pts Missing or incomplete solution.

- 0 pts All correct.

- 0 pts Syntax error. "!=" is invalid.

💬 y will be false when the days_attended > x and not >=

Question 9

Code output

4.5 / 5 pts

✓ + 1 pt Correct output for first print statement

✓ + 1 pt correct explanation of while loop

✓ + 1 pt correct number of times loop runs

✓ + 0.5 pts correct output value of first iteration of for loop

✓ + 0.5 pts correct explanation of first iteration of for loop

+ 0.5 pts correct output value of second iteration of for loop

✓ + 0.5 pts correct explanation of second iteration of for loop

+ 0 pts incorrect/unattempt

Question 10

Case analysis

2 / 6 pts

✓ + 0 pts Incorrect/Unattempted

+ 1 pt $x > 1$ as one of three parts of the **and** condition

+ 1 pt $y \leq x$ as one of three parts of the **and** condition

+ 1 pt $y \geq 1$ as one of three parts of the **and** condition

- 0.5 pts $x \geq 1$ (The student added an **extra equal sign** in the condition)

- 0.5 pts $y < x$ (The student **missed an equal sign** in one of the three conditions)

- 0.5 pts $y > 1$ (The student **missed an equal sign** in one of the three conditions)

- 1 pt Missing or extra cases specified

✓ + 1.5 pts Specified L7 as Line number

✓ + 1.5 pts Specified case 3 is unreachable

✓ - 1 pt Extra statement is added as unreachable



Question 11

Values of a and b

4 / 4 pts

+ 0 pts Unattempted

+ 0 pts Both are incorrect

✓ + 2 pts $b = 1$

✓ + 2 pts $a = 3 * LDIG$

LDIG = 0

Question 12

Expression in terms of n

2.5 / 5 pts

✓ + 1.5 pts If $n > 0$, then the loop executes **4 iterations**.

+ 1.5 pts If $n \leq 0$ then the loop executes **0 iterations**.

✓ + 1 pt If $n > 0$, the final output is **16n**.

+ 1 pt If $n \leq 0$, the final output is **n**.

+ 0 pts Unattempted or Wrong Answer

Question 13

Compute a list of lists

0 / 6 pts

Line 1:

valid range -> length of list2

+ 1 pt Correct

+ 0.5 pts Partially Correct

+ 0 pts Incorrect / Unattempted

Line 2: Either is correct

1. Initializing a placeholder for storing elements
2. Initializing a variable to track previous element
3. Storing value of current list2 element in a variable
4. Adding an empty list to list3

+ 1 pt Correct

+ 0.5 pts Partially Correct

+ 0 pts Incorrect / Unattempted

Line 3: Either is correct

1. Initializing a placeholder for storing elements
2. Initializing a variable to track previous element
3. Storing value of current list2 element in a variable
4. Adding an empty list to list3

+ 1 pt Correct

+ 0.5 pts Partially Correct

+ 0 pts Incorrect / Unattempted

Line 4:

Check for divisibility and uniqueness

+ 1 pt Correct

+ 0.5 pts Partially Correct

+ 0 pts Incorrect / Unattempted

Line 5: Either is correct

1. Add items to appropriate list in list3
2. Updating value of previous element

+ 1 pt Correct

+ 0.5 pts Partially Correct

+ 0 pts Incorrect / Unattempted

Line 6: Either is correct

1. Add items to appropriate list in list3

2. Updating value of previous element

+ 1 pt Correct

+ 0.5 pts Partially Correct

+ 0 pts Incorrect / Unattempted

✓ + 0 pts Complete solution Incorrect / Unattempted

Question 14

List different possible outputs

4.5 / 8 pts

Case1: non-integer

+ 0.5 pts output: ValueError: invalid literal for int() with base 10

✓ + 0.5 pts case: When R is a non-integer(string,float,char,etc)

Case2: 0 or 1

+ 1 pt case:if R=0, then $R//2 = 0$. output: ValueError: range() arg 3 must not be zero

✓ + 1 pt case:if R=1, then $R//2 = 0$. output: ValueError: range() arg 3 must not be zero

Case3: if R is odd,negative or 2.

✓ + 1 pt case:if R is odd strictly greater than 1. output: 1

✓ + 1 pt case:if R is negative: output: 1

+ 1 pt case:if R is 2. output: 1

Case4: if R is even and strictly greater than 2

✓ + 1 pt output: None

+ 1 pt case: $R \geq 4$ (positive integer greater than equal to 4)

+ 0 pts unanswered or wrong answer.

5 for R = 2 output will be 1. but you have said $R \geq 2$, R = even then output will be null which is partially correct

6 if entered value is not integer then it will throw an ValueError

7 for R = 1 it will throw an ValueError

Question 15

Finding logical error in program

4 / 5 pts

+ 0 pts Incorrect / Unattempted

✓ + 0.5 pts Mentioned Error : **factorial initialized as 0** → causes sum to reset incorrectly and factorial logic to break.

✓ + 0.5 pts Fix : L3 => `factorial = 1`

✓ + 1.5 pts Mentioned Error :
1) **Loop i ranges start at 0** → should start at 1 (currently includes extra/unwanted iteration) to n+1
2) **Loop j ranges over i** → should start from 0 to i + 1

✓ + 1.5 pts Fix : L4 => `for i in range(1, n+1)` or
Fix : L6 => `for j in range(i+1)` or
any other / similar way to fix the range issue. (But see the next rubric in case it is not fully correct.)

– 0.5 pts Incorrect Initialization of for loop. correct initialization is
`for i in range(1, n+1)` or
`for j in range(i+1)`

+ 0.5 pts Mentioned Error : **Missing else after if** → code runs for negative numbers too, printing factorial as 0

+ 0.5 pts Fix : **before L3** => `else :` or
any other method used to make sure code for negative number is executed correctly

+ 0 pts if every fix done without/with mentioning the errors (Ignore this Rubric)

Question 16

Complete the program

3 / 7 pts

+ 0.5 pts Realize the need for an extra place in L, but not creating it correctly

+ 1 pt Valid initialization (creation of a new place at the end of L)

Valid range

+ 1 pt Valid start index

✓ + 1 pt Valid end index

✓ + 1 pt Valid step size or default step size

Correct assignment statement for moving elements iteratively

+ 1.5 pts Correct value entered in left blank

+ 1.5 pts Correct value entered in right blank

+ 0 pts Incorrect/ not attempted

💬 + 1 pt The intent seems to be correct, using i instead of j in line 3.

Question 17

Output of program

5 / 7 pts

+ 7 pts Completely correct

Correctly print list1

✓ + 0.5 pts Recognize it as a list of lists

✓ + 0.5 pts Recognizing there are 5 inner lists

✓ + 1 pt For correct elements

Prints in the middle

✓ + 1 pt Applied inner break correctly :- For the sublist [0,1,4], the inner loop breaks after adding 4 in sum, which results in the output 0#4

✓ + 1 pt Applied inner break correctly :- all elements are correctly printed by the inner loops

✓ + 1 pt Applied outer break correctly :- Break terminates outer loop before iterating the last inner list (4 ',' commas must be in total)

+ 1 pt Applied outer break correctly :- All elements are correctly printed by the outer loops

+ 1 pt Correctly printing the final sum

- 0.5 pts '#' and ','(comma) are misplaced or absent

+ 0 pts Incorrect/Unattempted

COL1000 Mid-term Exam, 2025-26 Sem I (89 marks, 120 minutes)

Kerberos ID

CS1 251160

Name

Krishna Jaiswal

Re-enter the last digit of your Kerberos ID:

0

→ (This is your LDIG. You will need to use it in some questions below.)

Read all instructions and questions thoroughly and carefully. No clarifications may be sought during the exam. Write your kerberos ID on every sheet.

Please sign the pledge below: (Exam will not be graded without your signature)

I will neither seek help with the answers nor provide help during this exam.

KAT Jaiswal

Note: There are 17 questions on 4 pages. [] lists marks available for each question.

Limit your answers to given spaces. Rough-sheet is included in the back. Do not detach it.

If a question seems ambiguous, write down your interpretations, and answer all variants.

1. [4] You have to visit a website via a browser, but it throws up the following error. Explain the error. Analyze the listed actions – what set of steps should you consider and under what condition?

Someone could be trying to impersonate the site. You should not continue.

Websites prove their identity via certificates. This browser does not trust mybankk.com because its certificate issuer is unknown.

Error code: SEC_ERROR_UNKNOWN_ISSUER

Go Back (Recommended)

Accept the Risk and Continue

1. Meaning of the error code: The site may have threat for us

2. Action: Go back and check the certificate

3. Action: Continue and Accept the risk

2. [5] Given the following program, provide the requested values after the execution of each statement. Be sure to format lists and tuples appropriately.

x = LDIG

x += x

x, y = x+9/2, x+9//2

[lxy, txy] = list((x, y)), (x,y)

lxy2, txy2 = lxy, txy

lxy += lxy2

txy += txy2

Value of:

x	0
x	4.5
lxy	[4.5, 4]
lxy2	[4.5, 4]
lxy	[4.5, 4, 4.5, 4]
txy	(4.5, 4)

Value of:

y	4
txy	(4.5, 4)
txy2	(4.5, 4)
lxy2	[4.5, 4]
txy2	(4.5, 4)

3. [4] What is the value of each of the following expressions? If the expression has an error, specify that error instead.

a)	True == False	→ False
b)	True == not False	→ True
c)	not True == False	→ True
d)	10 == 5+5 > False	→ True

4. [5] Given the following program, provide the requested values after the execution of each statement. Do remember to format lists and tuples properly.

t1 = t2 = (0, 1, -LDIG)	# Initialize t1 and t2	
l1 = l2 = [0, t1, LDIG]	# Initialize l1 and l2	
t1 += t1[1:3]	→	t1 (0, 1, 0, 1, 0)
l1 = l1[1:3]	→	l1 [(0, 1, 0, 1, 0), 0]
l1.append(l2)	→	l1 [(0, 1, 0, 1, 0), 0, (0, 1, 0, 1, 0), 0]
l2 = [x*x for x in t2]	→	l2 [0, 1, 0]
l2 = [x*x for x in t2 if x != 0]	→	l2 [1]
l2 = [x*x if x > 0 else -x*x for x in t2 if x != 0]	→	l2 [1]

5. [3] What is printed in each of the following programs? If there is an error, specify the error instead.

a) l1, l1[0] = [(1,2,3), LDIG] print(l1)	l1 (1,2,3)
b) l1, l1[0] = [(1,2,3), LDIG] print(l1)	l1 [1,2,3]
c) name = 'Beta' if name == 'alpha' or 'Alpha': name = 'ALPHA' print(name)	name: it should be name == 'alpha' or name == 'Alpha':

6. [5] Consider each line of the following program in sequence. If there is an error on that line, given the earlier valid statements, specify the error. If there is no error due to that statement, write **None** instead.

value = [1]	→	None
2xvalue = value + [2]	→	Type Error: Cannot multiply int with a list
value = Value+[3]	→	Name Error: Name 'Value' is not defined
value[1]	→	Index error: Index out of range value = [1]
value[0], value[0] = 3	→	None

7. [4] What is the result of executing each of the following programs? Specify the error instead, if there is one. Denote the space character by @ and end-of-line by \$ in your output. Assume that the user input on prompt is 1,2 followed by End-of-line (i.e., Enter) each time.

a) input1 = input('input: ') print('user' + input1)	→	user 1,2 \$
b) input1 = input('input: ') print('user', input1)	→	user @ 1,2 \$
c) input1 = input('input: ') print('user', input1.split())	→	user @ 1 @ 2 \$
d) input1 = input('input: ') print('user' + input1.split())	→	user 1 @ 2 \$

8. [6] Assume **days_attended** refers to the number of classes attended and **total_days** refers to the total number of classes held. **Mark** refers to the total marks obtained by some student. The Table below maps marks to grade, except if **days_attended** is less than or equal to 3/4th of **total_days**: the listed grade must be lowered by 1 (subject to 0 remaining the lowest possible grade). Write Python code that computes the grade **Grade** for some student, given **Mark**, **days_attended**, and **total_days**. (You may write the code in two columns, left half first.)

marks ≥ 85	grade = 10
85 > marks ≥ 70	grade = 8
70 > marks ≥ 55	grade = 6
55 > marks ≥ 40	grade = 4
40 > marks	grade = 0

Assumption: Since not mentioned
total_days = 100
(We can also take input from user)

<pre>total_days = 100 x = (3/4) * total_days if days_attended >= x: y = False else: y = True # Defining a temp variable if Mark >= 85: Grade = 10 elif 85 > Mark >= 70: Grade = 8 elif 70 > Mark >= 55: Grade = 6</pre>	<pre>elif 55 >= Mark >= 40: Grade = 4 else: Grade = 0 if y = True and Grade > 0: Grade -= 1 print(Grade)</pre>
---	---

9. [5] What is printed by the following code? Explain.

```
ii = 5
while ii >= 1:
    jj = ii
    ii -= 1
    print(jj)
    for jj in range(jj, 5, 2):
        print(5+jj//2)
```

Output:

1
5
5

Explain:

- a) For first output, print statement is out of ~~box~~ loop, so it continues until ii = 0
b) For jj in range (1, 5, 2) it iterates for value 1, 3 But value of jj is 1

10. [6] Given int variables x and y, the following code categorizes cases. Does it cover all cases of possible values of x and y? If not, list the cases that are not covered. Does it have code that would never be executed (unreachable code)? If so, provide the line number(s) and the specific part(s) of the line(s)?

Assumption
L₁ is on L₂
case = 3
if x+y > 3
(else 4)
no sense

```
L1: if (x < -1 or y < -2):
L2:     case = 0
L3: elif (x > 1 or y > 2):
L4:     if (x < y): case = 1
L5:     elif y < 1: case = 2
L6: else:
L7:     case = 3 if x+y > 3 else 4
```

Cases Not covered:

For $x \in [-1, 1]$ and $y \in [-2, 2]$

Unreachable code:

L₄: else:

L₇: case 3 if x+y > 3

As for cases not covered
Sum (x+y) $\in [-3, 3]$

11. [4] a = LDIG
b = 5
while b > 1+b//2:
 a += LDIG
 b -= 2

Examine the program on the left.

→ What values do a and b refer to at the end?

a:

b:

- 1) a = 0
2) a = 0 b = 5
3) a = 0 b = 5, 1+b//2 = 3
4) a = 0 b = 5,
5) a = 0, b = 3
6) a = 0, b = 3, 1+b//2 = 2
7) a = 0, b = 3
8) a = 0 b = 1
9) a = 0, b = 1, 1+b//2 = 1 → Exits out of while loop

12. [5] nxt = n
while 2*nxt < 20*n:
 nxt += nxt
print(nxt)

Examine the program on the left, and provide the data requested below as an expression in n. Assume n is int.

The number of times the loop iterates:

The value printed at the end:

- a) nxt = n n = n
b) $2n < 20n \Rightarrow \text{nxt} = 2n$
c) $4n < 20n \Rightarrow \text{nxt} = 4n$
d) $8n < 20n \Rightarrow \text{nxt} = 8n$

- e) $\text{nxt} = 8n$
 $2*\text{nxt} < 20n$
 $\text{nxt} = 16n$
f) $2*\text{nxt} > 20n$
Do not perform while loop

13. [6] Given two lists of ints, namely list1 and list2, complete the following nested loop to compute a list of lists list3 such that list3[i] is the list of unique elements of list1, which are divisible by list2[i].

```
list1.sort() # sort list1 in increasing order
list3 = []
for i in range(0, len(list1), 1):
```

Example:

```
list1: [10,20,20,35] list2: [2,5]
→ list3: [[10, 20], [10, 20, 35]]
```

```
    for e in list1:
        if e % list2[j] == 0:
            list3.append(e)
            j += 1
```

For elements of list1
Condition that e may be added to list3
Add to appropriate list in list3 (Indent properly)

14. [8] Examine the given program. What are the possible output, considering all input cases, including any corner cases? List the different possible output and all input which result in that output.

```
R = int(input('R '))
Done = None
for r in range(R, 2*R, R/2):
    if r >= 2*R-1:
        Done = 1
if(Done): print(Done)
```

(In case there is no output, list **None** in the output box. If there is an error instead, specify that error. In the Case box, list the corresponding input.)

Output	Type Error	Case I	If number is not entered
Output	1	Case II	If $R < 0$ and $(R > 0, R = \text{odd})$
Output	None	Case III	If $(R > 0, R = \text{even})$
Output	Index error	Case IV	$R = 1 \Rightarrow R // 2 = 0$

15. [5] Point out the logical error(s) in the following program, which seeks to compute the factorial of an integer without using the * operator. Also fix those error(s). You may refer to the line numbers shown.

```
L1: n = int(input('Input an integer: ')) # Assume user enters an integer
L2: if n < 0: print('Factorial of negative integers are not defined')
L3: factorial = 0
L4: for i in range(n):
L5:     Sum = 0
L6:     for j in range(i):
L7:         Sum += factorial
L8:     factorial = Sum
L9: print(f'Factorial of {n} is {factorial}')
```

Errors: No matter how many times we iterate $\text{sum} = 0, \text{factorial} = 0$

Fix: $L_3: \text{factorial} = 1$
 $L_6: \text{for } j \text{ in range}(i+1):$

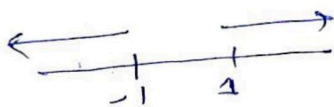
16. [7] Complete the following program to insert element e at index i of list L. Before insertion, each element at position i and greater must be shifted one position higher (i.e., $L[j]$ is moved to $L[j+1]$) to make space for e. Make sure that no elements of L are lost in the shifting process.

```
L = [0, 1, 2, ..., n] (n, e, i) # Any initialization
for j in range(i, len(L)): # Iterate over positions of the list
    L[j+1] = L[j] # Move element of L to make space
L[i] = e
```

17. [7] Provide the output of the following program.

```
list1 = [[i*i for i in range(j)] for j in range(5)]
print(list1)
sum = 0
for le in list1:
    for i in range(1, len(le)):
        if le[i] - le[i-1] > 1:
            sum += le[i]
            break
    print(sum, end='#')
print(sum, end=',')
if sum > 1:
    break
print(f'\nSum is {sum}')
```

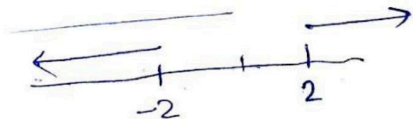
```
[[], [0], [0, 1], [0, 1, 4], [0, 1, 4, 9]]
0, 0, 0# 0, 0# 3,
Sum is 3
```



$$q = 0$$

$$1+2 =$$

$$q \pm = 0$$



$$t_1 = t_2 = (0, 1, 0)$$

$$l_1 = l_2 = [0, (0, 1, 0), 0]$$

$$n = 3$$



n

next

$2n$

$4n$

$8n$

$16n$

$20n$

$i=0$

$j=0$

$i=1$

$j=0$

$j=1$

$20n$

$$(0, 1, 0)$$

$$x < -1$$

$$y \checkmark$$

$$x > 1$$

$$y \checkmark$$

$$-1 < x < +1$$

$$-2 < y < 2$$

$$-3 < < 3$$



9) ii 5
 jj

~~[0, 1, 2, 3, 4]~~

~~jj~~

[], [0], [0, 1], [0, 1, 4], [0, 1, 4, 9]
 $j=0$ $j=1$ $j=2$ $j=3$ $j=4$

$R < 0$

(1, 2, 0)

(0, 0, 0)

1 2 0
 -1 -2 -1
 -2 -4 -1

2, 4, 1

2, 3

$le[1] - le[0]$

(-1)

$le[2] - le[1]$

-2

-3

$R=2$

(2, 4, 1)

2, 3

for i

x

(-5)

$2R - 1 = 3$

$R=3$

(3, 8, 1)

(3, 5)

5

[]

[0]

[0, 1]

[0, 1, 4]

(4, 8, 2)

$2 \times 4 - 1 = 7$

(5, 10, 2)

(5, 7, 3)

3

7, 14, 3

6, 12, 3

(2x6-1)

7, 10, 13

$1 \times 2 \times 3 \times 4$

(1)

(1+1)

(2+2+2)

6 + 6 + 6 + 6

$n=3$

Sum

Fact

0

1

2

3

(1)

(2)

$n=0$

$n=1$

$i=0$

~~$i=0$~~

$i=1$

$j=0$

(sum=1)

$i=0$

$j=0$

Sum=1

(2)

$n=2$

$i=0$

$i=1$

$j=0$

sum=2

$i=1$

$j=0$

Sum=2

(1)

$i=2$

$j=0$

$j=1$

$j=2$

(1+1)

$i=2$

$j=0$

Sum=3

$j=1$

Sum=5

$i=3$

$j=0$

$j=1$

$j=2$

1

2

(1 1)

24

6

6

6

6

2

2

2

1

1

$n=4$

$i=3$

$\rightarrow 6 + 6 + 6 + 6$

$i=2$

$\rightarrow 2 + 2 + 2$

$i=1$

$\rightarrow 1 + 1$

$i=0$

\rightarrow

~~(1 + n + 1)~~

$n=2$

(1+1)

$n=3$

(2+2+2)

1+1