

# Advanced Python Programming (4321602) - Summer 2023 Solution

Milav Dabgar

July 31, 2023

## Question 1

### પ્રશ્ન 1(a) [3 ગુણ]

લિસ્ટ શું છે? તેનો પાયથનમાં ઉપયોગ શું છે અને તેની લાક્ષણિકતાઓ લખો.

#### જવાબ

List એ items (elements) નું ordered collection છે જે એક જ વેર્બિલ માં multiple values store કરી શકે છે. લિસ્ટ mutable છે અને duplicate elements ની મંજૂરી આપે છે.

લાક્ષણિકતાઓ:

ફીચર	વર્ણન
Ordered	Elements નો કમ નિર્ધારિત હોય છે
Mutable	બનાવ્યા પછી બદલી શકાય છે
Indexed	Index [0,1,2...] વાપરીને access કરી શકાય
Duplicates	Duplicate values ની મંજૂરી છે

પાયથનમાં ઉપયોગ:

- Data Storage:** સંબંધિત items નો સંગ્રહ.
- Dynamic Arrays:** Runtime દરમિયાન size બદલી શકાય.
- Iteration:** Elements માં આસાનીથી loop કરી શકાય.

#### મેમરી ટ્રીક

OMID - Ordered, Mutable, Indexed, Duplicates

### પ્રશ્ન 1(b) [4 ગુણ]

પાયથનમાં String built-in functions સમજાવો.

#### જવાબ

String built-in functions પાયથન પ્રોગ્રામમાં text data ને efficiently manipulate અને process કરવામાં મદદ કરે છે.

સામાન્ય String Functions:

Function	હેતુ	ઉદાહરણ
upper()	Uppercase માં convert કરે	"hello".upper() → "HELLO"
lower()	Lowercase માં convert કરે	"WORLD".lower() → "world"
strip()	Whitespace remove કરે	" hi ".strip() → "hi"
split()	List માં split કરે	"a,b".split(",") → ['a', 'b']
replace()	Substring replace કરે	"cat".replace("c","b") → "bat"
find()	Substring position શોધો	"hello".find("e") → 1

#### મુખ્ય મુદ્દાઓ:

- **Immutable:** Original string અપરિવર્તિત રહે છે.
- **Return Values:** Functions નવી strings return કરે છે.
- **Case Sensitive:** Functions case ને ધ્યાનમાં રાખે છે.

#### મેમરી ટ્રીક

ULSR-FR - Upper, Lower, Strip, Replace, Find, Replace

## પ્રશ્ન 1(c) [7 ગુણ]

સેટમાંથી કોઈ element કેવી રીતે ઉમેરતું, દૂર કરવું તે લખો. POP remove થી કઈ રીતે અલગ છે તે સમજાવો.

#### જવાબ

Sets એ unique elements નો unordered collection છે.

#### Set Operations:

Operation	Method	Syntax	ઉદાહરણ
Add	add()	set.add(e)	s.add(5)
Remove	remove()	set.remove(e)	s.remove(3)
Safe Remove	discard()	set.discard(e)	s.discard(7)
Pop	pop()	set.pop()	s.pop()

#### Code ઉદાહરણ:

```

1 my_set = {1, 2, 3}
2 my_set.add(5)      # Add
3 my_set.remove(2)   # Remove specific
4 element = my_set.pop() # Remove random
5

```

#### POP vs REMOVE તફાવત:

પાસું	pop()	remove()
Target	Random element	Specific element
Parameter	જરૂરી નથી	Element value જરૂરી
Return	Removed element	None
Error	Set empty હોય તો error	Element ન મળે તો error

#### મેમરી ટ્રીક

PRRE - Pop Random, Remove Exact

## પ્રશ્ન 1(c) OR [7 ગુણ]

બિલ્ટ-ઇન Dictionary functions ની ચાદી લખો. Dictionary ના functions અને operations દર્શાવવા માટે પ્રોગ્રામ લખો.

### જવાબ

#### Dictionary Functions:

Function	હેતુ	Return કરે છે
keys()	બધી keys મેળવે	dict_keys object
values()	બધી values મેળવે	dict_values object
items()	Key-value pairs મેળવે	dict_items object
get()	Safe value retrieval	Value અથવા None
pop()	Remove કરીને value return કરે	Removed value
clear()	બધી items remove કરે	None
update()	Dictionaries merge કરે	None

#### પ્રોગ્રામ ઉદાહરણ:

```

1 student = {'name': 'John', 'age': 20}

2

3 # Operations
4 print(list(student.keys())) # Keys
5 print(student.get('age')) # Safe get
6 student.update({'grade': 'A'}) # Update
7 student.pop('age') # Remove
8

```

### મેમરી ટ્રીક

KVIGPCU - Keys, Values, Items, Get, Pop, Clear, Update

## Question 2

### પ્રશ્ન 2(a) [3 ગુણ]

Tuple ની વ્યાખ્યા લખો અને તે કઈ રીતે પાયથનમાં બનાવાય?

### જવાબ

Tuple એ ordered collection છે જે immutable છે (બનાવ્યા પછી બદલી શકતી નથી).

#### Tuple Creation Methods:

Method	Syntax	ઉદાહરણ
Parentheses	(item1, item2)	(1, 2, 3)
No Parentheses	item1, item2	1, 2, 3
Single Item	(item,)	(5,)
Empty TC	()	()

## મેમરી ટ્રીક

IOI - Immutable, Ordered, Indexed

## પ્રશ્ન 2(b) [4 ગુણ]

Module ના ફાયદાઓ સમજાવો.

## જવાબ

Modules એ Python files છે જેમાં functions, classes અને variables હોય છે.

ફાયદાઓ:

ફાયદો	લાભ
Reusability	એક વાર લખો, ધ્રણી વાર વાપરો
Organization	Code ને logical units માં વિભાજિત કરે
Namespace	Naming conflicts ટાળે
Maintainability	Easy debugging અને updates

## મેમરી ટ્રીક

RONM - Reusability, Organization, Namespace, Maintainability

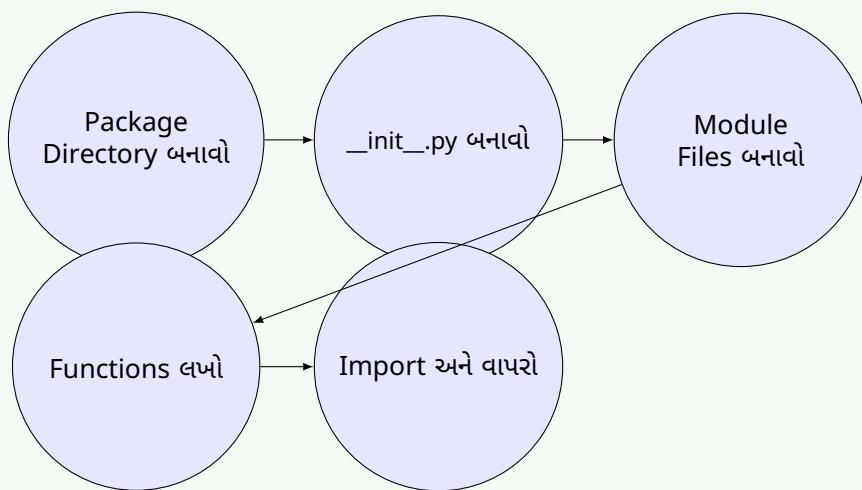
## પ્રશ્ન 2(c) [7 ગુણ]

થોડ્ય ઉદાહરણ સાથે user defined package બનાવવા માટેના steps લખો.

## જવાબ

Package એ directory છે જેમાં multiple modules અને \_\_init\_\_.py હોય છે.

Package બનાવવાના Steps:



## Step-by-Step Implementation:

1. Create Directory: mkdir mathtools
2. Create \_\_init\_\_.py:

```

1 # mathtools/__init__.py
2 print("Package loaded")
3

```

### 3. Create Module (basic.py):

```

1 def add(a, b): return a + b
2

```

### 4. Use Package:

```

1 import mathtools.basic
2 print(mathtools.basic.add(5, 3))
3

```

#### મેમરી ટ્રીક

DDMFU - Directory, Dunder-init, Modules, Functions, Use

## પ્રશ્ન 2(a) OR [3 ગુણ]

Tuple અને List વચ્ચેનો તફાવત લખો.

#### જવાબ

##### તફાવત:

ફીચર	Tuple	List
Mutability	Immutable	Mutable
Syntax	(1, 2, 3)	[1, 2, 3]
Performance	જડપી	ધીમું
Methods	મર્યાદિત	ઘર્ણી methods
Memory	ઓછી memory	વધુ memory

#### મેમરી ટ્રીક

TIF-LIM - Tuple Immutable Fixed, List Mutable Dynamic

## પ્રશ્ન 2(b) OR [4 ગુણ]

પાયથનમાં intra-package reference concept સમજાવો.

#### જવાબ

Intra-package references package અંદરના modules ને relative imports વાપરીને એકબીજાને import કરવાની મંજૂરી આપે છે.

##### Import પ્રકારો:

Type	Syntax	Usage
Absolute	from pkg.mod import fn	Full path
Relative	from .mod import fn	Same package
Parent	from ..mod import fn	Parent package

## મેમરી ટ્રીક

RAP - Relative, Absolute, Parent imports

## પ્રશ્ન 2(c) OR [7 ગુણ]

Module એટલે શું? વર્તુળનું ક્ષેત્રફળ અને પરિધ શોધવા માટે module બનાવવાનો પ્રોગ્રામ લખો. આ મ૗ડુલે ને પ્રોગ્રામમાં import કરો અને તેમાંથી functions call કરો.

## જવાબ

Module એ Python file છે જેમાં functions અને variables હોય છે.

## 1. Circle Module (circle.py):

```

1 import math
2
3 def area(r):
4     return math.pi * r * r
5
6 def circumference(r):
7     return 2 * math.pi * r
8

```

## 2. Main Program (main.py):

```

1 import circle
2
3 r = 5
4 print(f"Area: {circle.area(r):.2f}")
5 print(f"Circumference: {circle.circumference(r):.2f}")
6

```

## મેમરી ટ્રીક

IRUD - Import, Reuse, Use, Debug

## Question 3

## પ્રશ્ન 3(a) [3 ગુણ]

પાયથનમાં errors ના પ્રકારો સમજાવો.

### જવાબ

Errors ત્યારે આવે છે જ્યારે code properly execute ન થઈ શકે.

Error પ્રકારો:

Error Type	વર્ણન	ઉદાહરણ
Syntax Error	Code structure ખોટું	Colon, brackets ગુમ
Runtime Error	Execution દરમિયાન error	Zero થી division
Logical Error	ખોટું result	ખોટું formula

### મેમરી ટ્રીક

SRL - Syntax, Runtime, Logical

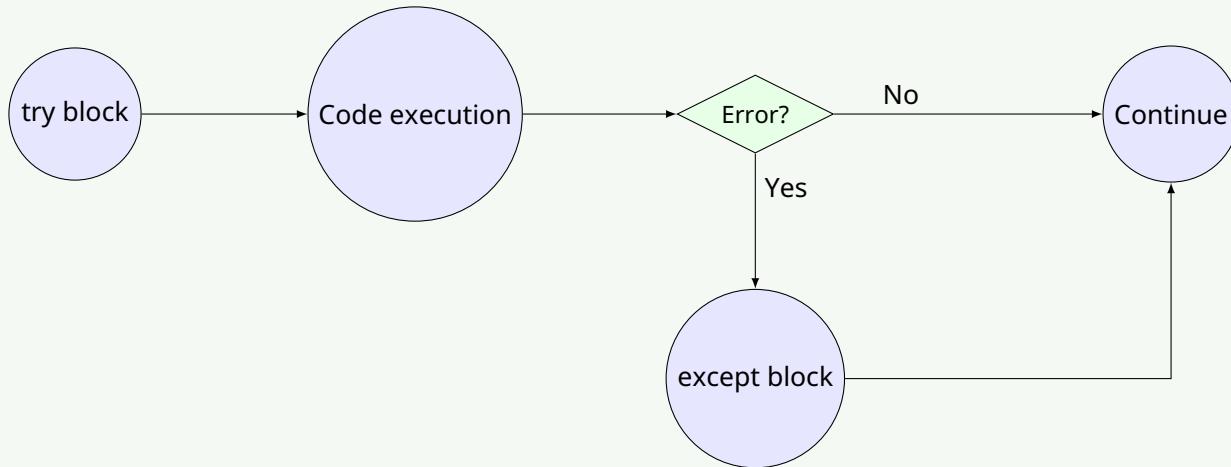
## પ્રશ્ન 3(b) [4 ગુણા]

try except નું structure સમજાવો.

### જવાબ

Try-except structure runtime errors ને gracefully handle કરે છે.

Basic Structure:



Syntax:

```

1 try:
2     # Risky code
3     code()
4 except ErrorType:
5     # Handle error
6     handle()
7 finally:
8     # Always execute
9     cleanup()
10
  
```

### મેમરી ટ્રીક

TEEF - Try, Except, Else, Finally

## પ્રશ્ન 3(c) [7 ગુણ]

Marks Result માટે એક function બનાવો જેમાં English અને Maths marks ની બે arguments હોય, જો કોઈપણ argument નું value 0 કરતાં ઓછું હોય તો error generate કરાવો.

### જવાબ

#### Implementation:

```

1 class InvalidMarksError(Exception):
2     def __init__(self, subject, marks):
3         super().__init__(f"Invalid {subject} marks: {marks}")
4
5     def marks_result(english, maths):
6         if english < 0: raise InvalidMarksError("English", english)
7         if maths < 0: raise InvalidMarksError("Mathematics", maths)
8
9         if english > 100: raise InvalidMarksError("English", english)
10        if maths > 100: raise InvalidMarksError("Mathematics", maths)
11
12        total = english + maths
13        percentage = (total / 200) * 100
14
15        if percentage >= 50: status = 'Pass'
16        else: status = 'Fail'
17
18        return {
19            'total': total,
20            'percentage': percentage,
21            'status': status
22        }
23

```

### મેમરી ટ્રીક

CVIR - Custom, Validate, Interactive, Robust

## પ્રશ્ન 3(a) OR [3 ગુણ]

પાયથનમાં built-in exceptions ની યાદી લખો (કોઈપણ પાંચ).

### જવાબ

#### Built-in Exceptions:

Exception	કરણ	ઉદાહરણ
ValueError	અમાન્ય મૂલ્ય	int("abc")
TypeError	ખોટો data type	"5"+5
IndexError	Index range બાદર	list[10]
KeyError	Key ન મળે	dict["x"]
ZeroDivisionError	Zero થી ભાગાકાર	10/0

## મેમરી ટ્રીક

VTIKZ - ValueError, TypeError, IndexError, KeyError, ZeroDivisionError

## પ્રશ્ન 3(b) OR [4 ગુણ]

`finally` પર મુદ્દા લખો અને ઉદાહરણ સાથે સમજાવો.

## જવાબ

**Finally Block:** exception આવે કે ન આવે, હંમેશા execute થાય છે.

લાક્ષણિકતાઓ:

- **Always Executes:** હંમેશા run થાય.
- **Cleanup:** Resources close કરવા માટે ઉત્તમ.

ઉદાહરણ:

```

1 try:
2     file = open("data.txt", "r")
3 except:
4     print("Error")
5 finally:
6     print("Cleanup")
7     file.close()
8

```

## મેમરી ટ્રીક

ARGC - Always Runs, Resource Cleanup

## પ્રશ્ન 3(c) OR [7 ગુણ]

Divide by Zero Exception ને `finally` clause સાથે catch કરતો પ્રોગ્રામ લખો.

## જવાબ

**Program:**

```

1 def safe_divide(a, b):
2     try:
3         print(f"Dividing {a} by {b}")
4         result = a / b
5         print(f"Result: {result}")
6     except ZeroDivisionError:
7         print("Error: Zero division not allowed!")
8     except TypeError:
9         print("Error: Numbers required!")
10    finally:
11        print("Operation completed")
12
13 safe_divide(10, 2)
14 safe_divide(5, 0)
15

```

## મેમરી ટ્રીક

CFLIS - Comprehensive, Finally, Logging, Interactive, Statistics

**Question 4****પ્રશ્ન 4(a) [3 ગુણા]**

File Handling શું છે? File Handling Operations ની ચારી આપો.

**જવાબ**

File Handling એ files ને read, write અને manipulate કરવાની પ્રક્રિયા છે.

Operations:

Operation	હેતુ	Method
Open	File ખોલવા	open()
Read	વાંચવા	read()
Write	લખવા	write()
Close	બંધ કરવા	close()

## મેમરી ટ્રીક

ORWCST - Open, Read, Write, Close, Seek, Tell

**પ્રશ્ન 4(b) [4 ગુણા]**

Object Serialization સમજાવો.

**જવાબ**

Object Serialization એ Python objects ને byte stream માં convert કરવાની પ્રક્રિયા છે.

Methods:

- Pickle: Python specific binary format.
- JSON: Text format, web માટે ઉપયોગી.

Pickle ઉદાહરણ:

```

1 import pickle
2 data = [1, 2, 3]
3 # Serialize
4 with open('data.pkl', 'wb') as f:
5     pickle.dump(data, f)
6

```

## મેમરી ટ્રીક

SPDT - Store, Persist, Data Transfer

## પ્રશ્ન 4(c) [7 ગુણ]

File માં રહેલા vowels ગણવાનો પ્રોગ્રામ લખો.

### જવાબ

#### Vowel Counter:

```

1 def count_vowels(filename):
2     vowels = 'aeiouAEIOU'
3     count = 0
4     try:
5         with open(filename, 'r') as f:
6             text = f.read()
7             for char in text:
8                 if char in vowels:
9                     count += 1
10            print(f"Total Vowels: {count}")
11        except FileNotFoundError:
12            print("File not found")
13
14 # Test
15 with open("test.txt", "w") as f:
16     f.write("Hello World")
17 count_vowels("test.txt")
18

```

### મેમરી ટ્રીક

FVESI - File Validation, Vowel Extraction, Statistics, Interactive

## પ્રશ્ન 4(a) OR [3 ગુણ]

File કેવી રીતે open અને close કરવી? તેની syntax આપો.

### જવાબ

**Modes:** 'r' (Read), 'w' (Write), 'a' (Append).

**Syntax:**

```

1 # Manual
2 f = open("file.txt", "r")
3 f.close()
4
5 # With statement (Recommended)
6 with open("file.txt", "r") as f:
7     data = f.read()
8

```

### મેમરી ટ્રીક

ORWA - Open, Read, Write, Append modes

## પ્રશ્ન 4(b) OR [4 ગુણ]

Text file અને Binary file વર્ચ્યોનો તફાવત લખો.

### જવાબ

#### તફાવત:

પાસું	Text File	Binary File
<b>Content</b>	Human readable chars	Bytes
<b>Mode</b>	'r', 'w'	'rb', 'wb'
<b>Encoding</b>	ASCII/UTF-8	None
<b>Size</b>	મોટી	નાની (compact)

#### મેમરી ટ્રીક

TCEB - Text Character Encoding Bigger, Binary Compact Efficient

## પ્રશ્ન 4(c) OR [7 ગુણ]

Seat no અને Name store કરવા માટે binary file બનાવવાનો પ્રોગ્રામ લખો. Seat no થી search કરી name display કરો.

### જવાબ

#### Program:

```

1 import pickle
2
3 # Add record
4 def add(seat, name):
5     record = {seat: name}
6     with open("student.dat", "wb") as f:
7         pickle.dump(record, f)
8
9 # Search
10 def search(seat):
11     try:
12         with open("student.dat", "rb") as f:
13             d = pickle.load(f)
14             if seat in d:
15                 print(f"Found: {d[seat]}")
16             else:
17                 print("Not found!")
18     except:
19         print("Error")
20
21 add(1, "Ram")
22 search(1)
23

```

#### મેમરી ટ્રીક

BSECH - Binary Storage, Search Efficiently, CRUD Handling

## Question 5

### પ્રશ્ન 5(a) [3 ગુણ]

Turtle શું છે અને objects draw કરવા માટે તેનો ઉપયોગ કેવી રીતે થાય છે?

#### જવાબ

Turtle એ Python graphics module છે જે drawing canvas અને cursor (turtle) પ્રદાન કરે છે.  
ઉદાહરણ:

```
1 import turtle
2 t = turtle.Turtle()
3 t.forward(100)
4
```

#### મેમરી ટ્રીક

CPTT - Canvas, Pen, Turtle, Teaching tool

### પ્રશ્ન 5(b) [4 ગુણ]

Turtle ને બીજી position પર move કરવા માટેની વિવિધ રીતો સમજાવો.

#### જવાબ

Movement Methods:

Method	કાર્ય
forward(d)	આગળ વધે
backward(d)	પાછળ જાય
goto(x,y)	(x,y) પર જાય
penup()	Drawing બંધ કરે
pendown()	Drawing શરૂ કરે

#### મેમરી ટ્રીક

FGPRS - Forward, Goto, Penup, Rotate, Set coordinates

### પ્રશ્ન 5(c) [7 ગુણ]

Turtle માં loops કેવી રીતે ઉપયોગી છે તે સમજાવો અને ઉદાહરણ આપો.

#### જવાબ

Loops નો ઉપયોગ patterns અને shapes ને efficiently draw કરવા માટે થાય છે.

ઉદાહરણ (Square Loop):

```
1 import turtle
2 t = turtle.Turtle()
```

```

3
4   for i in range(4):
5     t.forward(100)
6     t.right(90)
7

```

**મેમરી ટ્રીક**

LPDC - Loops, Patterns, DynamicGraphics, ComplexDesigns

**પ્રશ્ન 5(a) OR [3 ગુણ]****Turtle માં Shape function સમજાવો. કેટલા પ્રકારના shapes ઉપલબ્ધ છે?****જવાબ****Shape function cursor નો દેખાવ બદલે છે.****Shapes:** Arrow, Turtle, Circle, Square, Triangle, Classic.

```

1   t.shape("turtle")
2

```

**મેમરી ટ્રીક**

ATCSTC - Arrow, Turtle, Circle, Square, Triangle, Classic

**પ્રશ્ન 5(b) OR [4 ગુણ]****Turtle માં વિવિધ પ્રકારના pen commands સમજાવો.****જવાબ****Pen Commands:**

- **penup()**: Drawing અટકાવે.
- **pendown()**: Drawing શરૂ કરે.
- **pensize(w)**: Line જાડાઈ નક્કી કરે.
- **pencolor(c)**: Line કલર નક્કી કરે.

**પ્રશ્ન 5(c) OR [7 ગુણ]****Turtle નો ઉપયોગ કરીને ભારતીય ધવજ દોરવા માટેનો પ્રોગ્રામ લખો.****જવાબ****ભારતીય ધવજ Program:**

```

1 import turtle
2
3 def draw_rect(color, x, y, width, height):
4   t.penup()

```

```
5     t.goto(x, y)
6     t.pendown()
7     t.color(color)
8     t.begin_fill()
9     for _ in range(2):
10        t.forward(width)
11        t.right(90)
12        t.forward(height)
13        t.right(90)
14     t.end_fill()
15
16 t = turtle.Turtle()
17 t.speed(5)
18 width = 300
19 height = 50
20
21 # પટ્ટીઓ દોરો
22 draw_rect("orange", -150, 100, width, height)
23 draw_rect("white", -150, 50, width, height)
24 draw_rect("green", -150, 0, width, height)
25
26 # ચક્ર દોરો
27 t.penup()
28 t.goto(0, 0)
29 t.pendown()
30 t.color("navy")
31 t.circle(25)
32 # Spokes
33 for i in range(24):
34     t.penup()
35     t.goto(0, 25)
36     t.pendown()
37     t.forward(25)
38     t.backward(25)
39     t.right(15)
40
```

### મેમરી ટ્રીક

SWACP - Stripes, White-chakra, Accurate, Colors, Proportional