

# OOPS અને પાયથોન પ્રોગ્રામિંગ (4351108) - સમર 2025 સોલ્યુશન

Milav Dabgar

May 14, 2025

## પ્રશ્ન 1(a) [3 ગુણ]

Python માં for લૂપનો ઉદ્દેશ્ય શું છે? ઉદાહરણ સાથે સમજાવો.

### જવાબ

for લૂપનો ઉપયોગ કોઈ sequence (જેમ કે લિસ્ટ, ટપલ, સ્ટ્રિંગ) અથવા અન્ય iterable ઓફજેક્ટ પર પુનરાવર્તન કરવા માટે અને sequence ના દરેક આઇટમ માટે કોડનો બ્લોક ચલાવવા માટે થાય છે.

### કોડ ઉદાહરણ:

```
1 # દરેક ફળને લસ્ટમાંથી પ્રનિઃષ્ટ કરો
2 fruits = ["apple", "banana", "cherry"]
3 for fruit in fruits:
4     print(fruit)
```

- પુનરાવર્તન: સ્વયંસંચાલિત રીતે દરેક આઇટમ માટે કોડ પુનરાવર્તિત કરે છે
- સરળતા: કાઉન્ટર્સ સાથે while લૂપ્સ કરતાં સ્વચ્છ

### મેમરી ટ્રીક

"દરેક આઇટમ માટે કરો"

## પ્રશ્ન 1(b) [4 ગુણ]

Python માં variable ડિફાઇન કરવાના નિયમો જણાવો અને Python માં ડેટાપ્કારો (data types) ની યાદી આપો.

### જવાબ

વેરિએબલ ડિફાઇન કરવાના નિયમો:

કોષ્ટક 1. વેરિએબલ નિયમો

નિયમ	ઉદાહરણ	અમાન્ય ઉદાહરણ
અક્ષર અથવા અંડરસ્કોરથી શરૂ થયું જોઈએ	name = "John"	1name = "John"
અક્ષરો, નંબરો, અંડરસ્કોર સમાવિષ્ટ કરી શકે	user_1 = "Alice"	user-1 = "Alice"
કેસ-સેન્સિટિવ	age ≠ Age	-
રિઝર્વ કોર્ટીનો ઉપયોગ ન કરી શકાય	count = 5	if = 5

પાયથોન ડેટા ટાઈપ્સ:

કોષ્ટક 2. ડેટા ટાઈપ્સ

કેટા ટાઈપ	વિવરણ	ઉદાહરણ
int	પૂર્ણક સંખ્યાઓ	x = 10
float	દશાંશ સંખ્યાઓ	y = 10.5
str	ટેક્સ્ટ સ્ટિંગ્સ	name = "John"
bool	બૂલિયન મૂલ્યો	is_active = True
list	ક્રમબદ્ધ, બદલી શકાય તેવા	["apple", "banana"]
tuple	ક્રમબદ્ધ, બદલી ન શકાય તેવા	(10, 20)
dict	કી-વેલ્યુ જોડી	{"name": "John"}
set	અનોર્ડ્ડ અનન્ય	{1, 2, 3}

### મેમરી ટ્રીક

“SILB-DTS: String, Integer, List, Boolean, Dictionary, Tuple, Set”

## પ્રશ્ન 1(c) [7 ગુણ]

1 થી N સુધીના પ્રાઇમ નંબર પ્રિન્ટ કરવા પ્રોગ્રામ બનાવો.

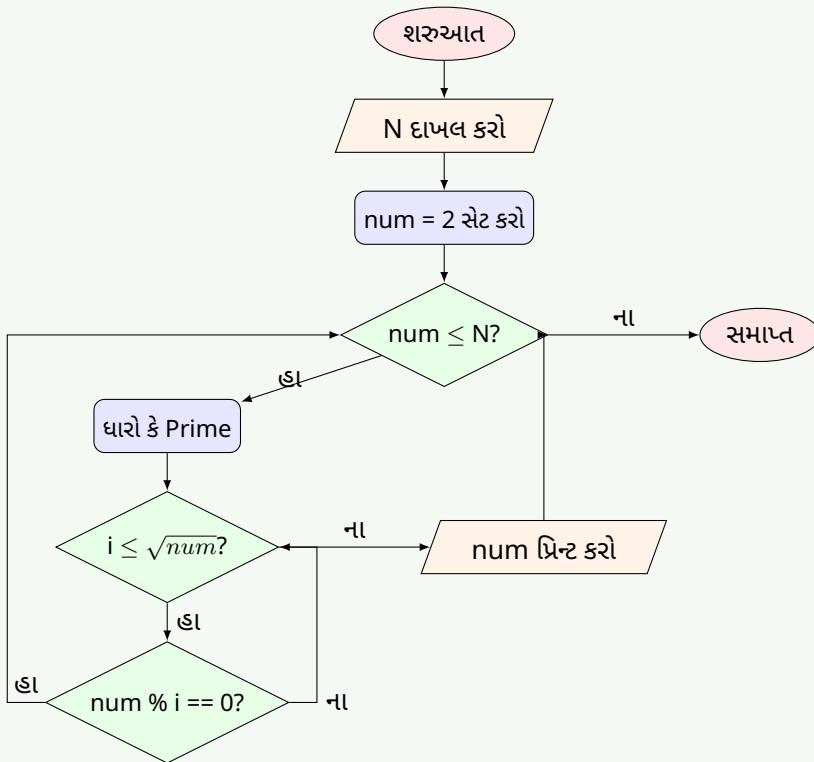
### જવાબ

```

1 def print_primes(n):
2     print("1 અને", n, "વચ્ચેના પ્રાઇમ નંબરો:")
3
4     for num in range(2, n + 1):
5         is_prime = True
6
7         # Check if num is divisible by any number from 2 to sqrt(num)
8         for i in range(2, int(num**0.5) + 1):
9             if num % i == 0:
10                 is_prime = False
11                 break
12
13             if is_prime:
14                 print(num, end=" ")
15
16     # Get input from user
17     N = int(input("N નંબર દાખલ કરો: "))
18     print_primes(N)

```

એલોરિધમ ડાયાગ્રામ:



આકૃતિ 1. Prime Number Algorithm

- ટાઇમ કોમ્પ્લેક્સિટી:  $O(N\sqrt{N})$
- સ્પેસ કોમ્પ્લેક્સિટી:  $O(1)$

### મેમરી ટ્રીક

“ભાગ કરીને પ્રાઇમ નક્કી કરો”

### પ્રશ્ન 1(c OR) [7 ગુણ]

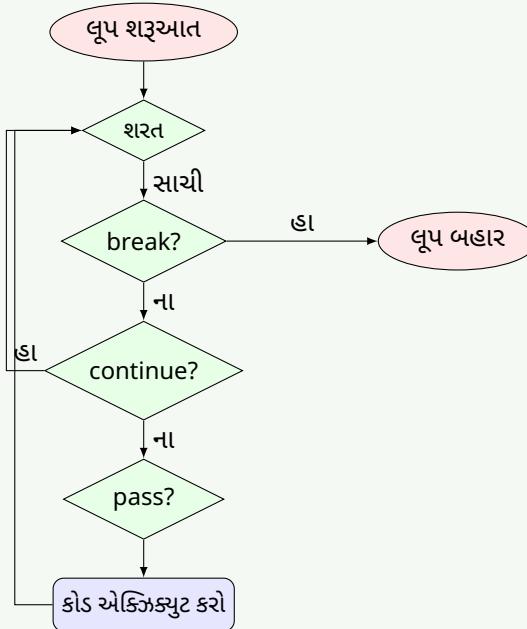
Python માં break, continue, અને pass સ્ટેમેન્ટનું કાર્ય અને ઉદાહરણ સાથે સમજાવો.

#### જવાબ

##### કોષ્ટક 3. કંટ્રોલ સ્ટેમેન્ટ્સ

સ્ટેમેન્ટ	ઉદ્દેશ	ઉદાહરણ
break	લૂપને સંપૂર્ણપણે સમાપ્ત કરે છે	લૂપ બંધ કરો
continue	આગલા પુનરાવર્તન પર જાય છે	આઈટમ્સ છોડો
pass	કંઈ કરતું નથી (પ્લેસહોલ્ડર)	ભવિષ્યનો કોડ

ફલો કંટ્રોલ ડાયગ્રામ:



આકૃતિ 2. Loop Control Logic

## ઉદાહરણ કોડ:

```

1 for i in range(5):
2     if i == 2: continue # Skip 2
3     if i == 4: break   # Stop at 4
4     if i == 0: pass    # Do nothing
5     print(i)
6 # Output: 0, 1, 3
  
```

## મેમરી ટ્રીક

“BCP - સંપૂર્ણપણે બંધ કરો, આંશિક રીતે ચાલુ રાખો, શાંતિથી પસાર થાઓ”

## પ્રશ્ન 2(a) [3 ગુણ]

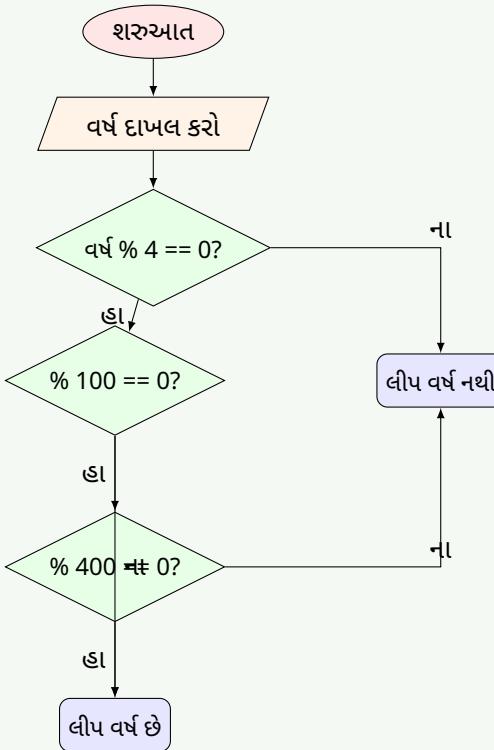
યુગરે આપેલ વર્ષ લીપ વર્ષ છે કે નહીં તે માટે પ્રોગ્રામ બનાવો.

## જવાબ

```

1 year = int(input("વર્ષ દાખલ કરો: "))
2
3 if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
4     print(f"{year} લીપ વર્ષ છે")
5 else:
6     print(f"{year} લીપ વર્ષ નથી")
  
```

## નિર્ણય વૃક્ષ:



આકૃતિ 3. Leap Year Logic

## મેમરી ટ્રીક

“4 હા, 100 ના, 400 હા”

## પ્રશ્ન 2(b) [4 ગુણ]

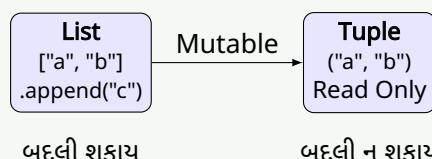
Python માં લિસ્ટ અને ટ્યુપલ વર્ચેના મુખ્ય તફાવત શું છે?

## જવાબ

કોષ્ટક 4. લિસ્ટ અને ટ્યુપલ

વિશેષતા	લિસ્ટ	ટ્યુપલ
સિન્ટેક્સ	[]	()
પરિવર્તનશીલતા	મ્યુટેબલ (બદલી શકાય)	ઇમ્યુટેબલ (બદલી ન શકાય)
પરોમન્સ	ધીમું	ઝડપી
ઉપયોગ	જ્યારે ફેરફાર જરૂરી હોય	જ્યારે ડેટા ફિક્સ હોય
મેમરી	વધુ મેમરી	ઓછી મેમરી

તુલના ડાયાગ્રામ:



આકૃતિ 4. List vs Tuple

## મેમરી ટ્રીક

“LIST - બદલી શકાય, TUPLE - બદલી ન શકાય”

## પ્રશ્ન 2(c) [7 ગુણ]

ચુંઝારે દાખલ કરેલ તમામ positive number છે કે નહીં તે શોધવાનો પ્રોગ્રામ બનાવો. જ્યારે ચુંઝારે negative number દાખલ કરે, ત્યારે ઇનપુટ લેવાનું બંધ કરો અને તમામ positive number નો સરવાળો કરો.

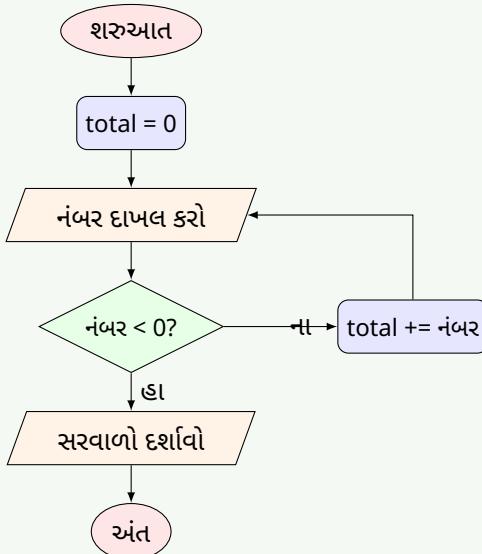
## જવાબ

```

1 def sum_positives():
2     total_sum = 0
3     while True:
4         num = float(input("નંબર દાખલ કરો (negative બંધ કરવા): "))
5         if num < 0:
6             break
7         total_sum += num
8     print(f"બધા પોઝિટિવ નંબરનો સરવાળો: {total_sum}")
9
10 sum_positives()

```

પ્રક્રિયા ફ્લો:



આકૃતિ 5. Summation Logic

## મેમરી ટ્રીક

“નેગેટિવ આવે ત્યાં સુધી સરવાળો કરો”

## પ્રશ્ન 2(a OR) [3 ગુણ]

તમે આપેલ ત્રણ નંબર માંથી મોટો નંબર શોધવાનું પ્રોગ્રામ બનાવો.

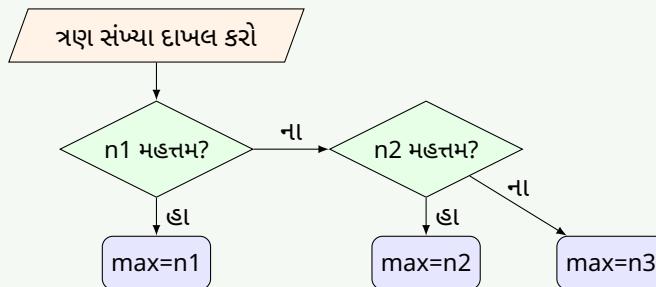
**જવાબ**

```

1 n1 = float(input("પહેલી સંખ્યા: "))
2 n2 = float(input("બીજી સંખ્યા: "))
3 n3 = float(input("તૃઠી સંખ્યા: "))

4
5 if n1 >= n2 and n1 >= n3:
6     mx = n1
7 elif n2 >= n1 and n2 >= n3:
8     mx = n2
9 else:
10    mx = n3
11
12 print(f"માહત્તમ સંખ્યા: {mx}")

```

**તુલના લોજિક:****આંકૃતિક 6. Maximum Finder Logic****મેમરી ટ્રીક****“દરેકની તુલના કરો, મોટામાં મોટો લો”****પ્રશ્ન 2(b OR) [4 ગુણ]**

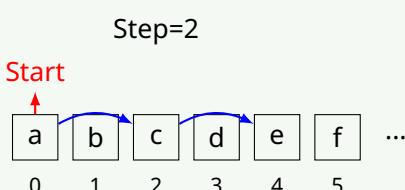
str = "abcdefghijklmnopqrstuvwxyz" આપેલ છે. ઉપરોક્ત સ્થિરમાંથી દરેક બીજાં અક્ષર જુદો કાઢવા માટે Python પ્રોગ્રામ લખો.

**જવાબ**

```

1 s = "abcdefghijklmnopqrstuvwxyz"
2 # સનિટેક્સ: [start:end:step]
3 result = s[0::2]
4 print("દરેક બીજી અક્ષર:", result)
5 # આઉટપુટ: acegikmoqsuwy

```

**સ્થિર સ્લાઇસિંગ ડાયાગ્રામ:****આંકૃતિક 7. String Slicing Step 2**

## મેમરી ટ્રીક

"સ્લાઇસ સ્ટેપ સિલેક્ટર"

## પ્રશ્ન 2(c OR) [7 ગુણ]

વિદ્યાર્થીઓના નામ અને તેમના માર્ક્સ સંગ્રહિત કરવા માટે ડિક્શનરી બનાવવાનું Python પ્રોગ્રામ લખો. 75 થી વધુ માર્ક્સ મેળવનાર વિદ્યાર્થીઓના નામ ડિસ્પ્લે કરવો.

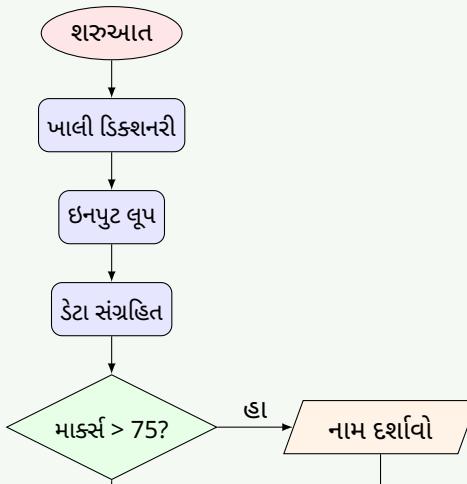
## જવાબ

```

1 students = {}
2 n = int(input("વિદ્યાર્થીઓની સંખ્યા: "))
3
4 # ઇનપુટ લૂપ
5 for i in range(n):
6     name = input("નામ: ")
7     marks = float(input("માર્ક્સ: "))
8     students[name] = marks
9
10 print("\n75 થી વધુ માર્ક્સ મેળવનારાં:")
11 for name, marks in students.items():
12     if marks > 75:
13         print(f"{name}: {marks}")

```

## પ્રક્રિયા ડાયાગ્રામ:



## મેમરી ટ્રીક

"બધું સંગ્રહો, કેટલાક ફિલ્ટર કરો"

## પ્રશ્ન 3(a) [3 ગુણ]

સ્પેસને બહાર રાખીને સ્ટ્રિંગની લંબાઈ શોધવાનો પ્રોગ્રામ લખો.

### જવાબ

```

1 s = input("સ્ટ્રિંગ દાખલ કરો: ")
2 no_spaces = s.replace(" ", "")
3 length = len(no_spaces)
4 print(f"સ્પેસ વનિા લંબાઈ: {length}")

```

વિચ્યુલાઇઝન:



આકૃતિ 9. Space Removal

### મેમરી ટ્રીક

“અક્ષરો ગણો, સ્પેસ છોડો”

## પ્રશ્ન 3(b) [4 ગુણ]

Python માં ડિક્ષનરી methods યાદી આપો અને દરેકને ચોગ્ય ઉદાહરણ સાથે સમજાવો.

### જવાબ

#### કોષ્ટક 5. ડિક્ષનરી મેથ્ડ્સ

મેથ્ડ	વિવરણ	ઉદાહરણ
get(k)	કી માટે મૂલ્ય મેળવો	d.get('a')
keys()	બધી કી મેળવો	list(d.keys())
values()	બધા મૂલ્યો મેળવો	list(d.values())
items()	(key, value) જોડી	d.items()
pop(k)	આઇટમ દૂર કરો	d.pop('a')
update()	ડિક્ષનરી અપડેટ કરો	d.update(d2)
clear()	ડિક્ષનરી ખાલી કરો	d.clear()

### મેમરી ટ્રીક

“GCUP-KPIV”

## પ્રશ્ન 3(c) [7 ગુણ]

Python ના લિસ્ટ ડેટા ટાઇપને સમજાવો.

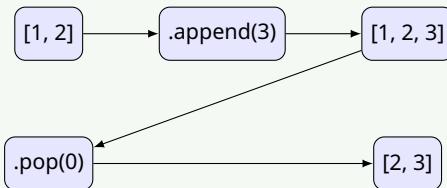
### જવાબ

લિસ્ટ કમબદ્ધ, પરિવર્તનશીલ સંગ્રહ છે જે ડુલિકેટ ઘટકો અને મિશ્ર પ્રકારોને મંજૂરી આપે છે.

મુખ્ય વિશેષતાઓ:

- કમબદ્ધ: આઇટમ્સ કમ જાળવી રાપે છે.
- પરિવર્તનશીલ: આઇટમ્સ ઉમેરી, દૂર કરી, બદલી શકાય છે.

- વિવિધ: int, str, float એક્સાથે સ્ટોર કરી શકાય છે.
- લિસ્ટ ઓપરેશન્સ ડાયાગ્રામ:



આકૃતિ 10. List Operations

## મેમરી ટ્રીક

“CAMP-IS: Create, Access, Modify, Process”

## પ્રશ્ન 3(a OR) [3 ગુણ]

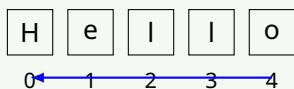
યુઝર પાસેથી સ્ટ્રિંગ ઇનપુટ લેવા માટેનું પ્રોગ્રામ લખો અને નવી સ્ટ્રિંગ બનાવ્યા વિના તેને reverse order માં છાપો.

## જવાબ

```

1 s = input("સ્ટ્રિંગ દાખલ કરો: ")
2 print(f"ઉલ્ટો: {s[::-1]}")
  
```

## રિવર્સિંગ લોજિક:



Step = -1 (પાછળની તરફ)

આકૃતિ 11. String Reversal

## મેમરી ટ્રીક

“પાછળની તરફ સ્લાઇસ કરો”

## પ્રશ્ન 3(b OR) [4 ગુણ]

Python માં ડિક્ષનરી ઓપરેશન્સની યાદી આપો અને દરેકને ઘોગ્ય ઉદાહરણ સાથે સમજાવો.

## જવાબ

કોષ્ટક 6. ડિક્ષનરી ઓપરેશન્સ

ઓપરેશન	કોડ	વિવરણ
એક્સેસ	d['key']	મૂલ્ય મેળવવા
ઉમેરો/બદલો	d['k'] = v	દાખલ/અપડેટ કરો
ડિલિટ	del d['k']	જોડો ફૂર કરો
ચકાસણી	'k' in d	મેમ્બરશિપ
લંબાઈ	len(d)	આઇટમ્સ ગણવા

## મેમરી ટ્રીક

``CADMIL: Create Access Delete Modify Iterate Length''

## પ્રશ્ન 3(c OR) [7 ગુણ]

Python ના સેટ ડેટા ટાઇપને વિગતે સમજાવો.

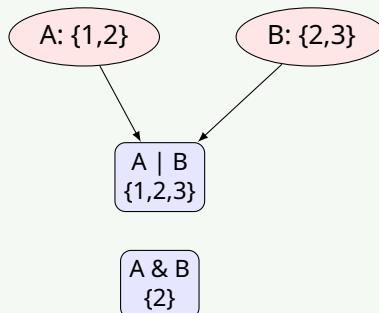
## જવાબ

સેટ અનન્ય ઘટકોનો અનોર્ડ્ડ સંગ્રહ છે.

સેટ લાક્ષણિકતાઓ:

- અનન્ય: કોઈ દૂસ્થિક્રિટસ નહીં.
- અનોર્ડ્ડ: કોઈ ઇન્ડક્સ એક્સેસ નહીં.
- ગાણિતિક: યુનિયન, ઇન્ટરસેક્શન સપોર્ટ કરે છે.

સેટ ઓપરેશન્સ ડાયાગ્રામ:



આકૃતિ 12. Set Union &amp; Intersection

## મેમરી ટ્રીક

``SUMO: Set Unique Mutable Ordered-less''

## પ્રશ્ન 4(a) [3 ગુણ]

statistics મોડ્યુલને સમજાવો અને તેમાંની ત્રણ પદ્ધતિઓ સાથે ઉદાહરણ આપો.

## જવાબ

## કોષ્ટક 7. Statistics Methods

મેથડ	વિવરણ	ઉદાહરણ
mean()	સરેરાશ	mean([1,2,3]) → 2
median()	મધ્ય મૂલ્ય	median([1,5,9]) → 5
mode()	સૌથી સામાન્ય	mode([1,1,2]) → 1

## મેમરી ટ્રીક

``MMM Stats''

## પ્રશ્ન 4(b) [4 ગુણ]

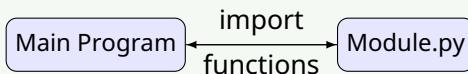
Python માં યુઝર ડિફાઇન્ડ ફંક્શન અને યુઝર ડિફાઇન્ડ મોડ્યુલને સમજાવો.

## જવાબ

કોષ્ટક 8. ફંક્શન અને મોડ્યુલ

વિશેષતા	ફંક્શન	મોડ્યુલ
એકમ	કોડ બ્લોક	ફાઇલ (.py)
નિર્માણ	def name():	.py તરીકે સાચવો
ઉપયોગ	name() કોલ	import name
સ્કોપ	લોકલ	ગ્લોબલ/ઇમ્પોર્ટ્સ

મોડ્યુલ માળખું:



આકૃતિ 13. Import Relationship

## મેમરી ટ્રીક

``FIR-MID''

## પ્રશ્ન 4(c) [7 ગુણ]

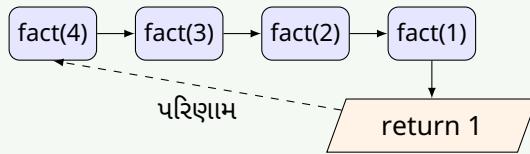
Using recursion આપેલ આંકડાના ફેક્ટોરીયલને શોધવા માટે યુઝર ડિફાઇન્ડ ફંક્શનનો ઉપયોગ કરીને Python કોડ લખો.

## જવાબ

```

1 def factorial(n):
2     # વેજ કેસ
3     if n == 0 or n == 1:
4         return 1
5     # રક્રસિવ કેસ
6     else:
7         return n * factorial(n-1)
8
9 num = int(input("નંબર દાખલ કરો: "))
10 print(f"ફેક્ટોરીયલ: {factorial(num)}")
  
```

રિકર્ઝન વિજ્ઞુલાઇઝેશન:



આકૃતિ 14. Recursion Chain

મેમરી ટ્રીક

“સંખ્યા ગુણ્યા (સંખ્યા માઇનસ વન)!“

## પ્રશ્ન 4(a OR) [3 ગુણ]

મેથ મોડ્યુલને સમજાવો અને તેમાંની ત્રણ મેથ્ડ્સ ઉદાહરણ સાથે સમજાવો.

જવાબ

કોષ્ટક 9. Math Methods

મેથ્ડ	વિવરણ	ઉદાહરણ
sqrt()	વર્ગમૂળ	sqrt(16) → 4.0
pow()	પાવર	pow(2,3) → 8.0
ceil()	ઉપર રાઉન્ડ	ceil(4.1) → 5

મેમરી ટ્રીક

“SPT Math”

## પ્રશ્ન 4(b OR) [4 ગુણ]

Python માં global અને local variables સમજાવો.

જવાબ

કોષ્ટક 10. વેરિએબલ સ્કોપ

પ્રકાર	સ્કોપ	એક્સેસ
ગલોબલ	આખો પ્રોગ્રામ	ગમે ત્યાં
લોકલ	ફંક્શનની અંદર	માત્ર ફંક્શનમાં

સ્કોપ ડાયાગ્રામ:

## Global Scope

Function\_scope

local\_var

આકૃતિ 15. Scope Hierarchy

## મેમરી ટ્રીક

"GLOBAL બધે જાય, LOCAL ફક્ત ફંક્શનમાં રહે"

## પ્રશ્ન 4(c OR) [7 ગુણ]

આપેલ સ્ટ્રિંગ પેલિન્ડ્રોમ છે કે નહીં તે તપાસવા માટે યુગર ડિફાઇન્ડ ફંક્શન બનાવો.

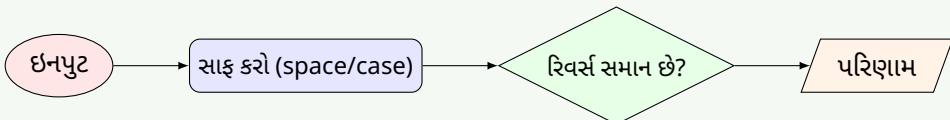
## જવાબ

```

1 def is_palindrome(text):
2     raw = text.replace(" ", "").lower()
3     return raw == raw[::-1]
4
5 s = input("સ્ટ્રિંગ દાખલ કરો: ")
6 if is_palindrome(s):
7     print("પેલિન્ડ્રોમ છે")
8 else:
9     print("પેલિન્ડ્રોમ નથી")

```

## લોજિક ફ્લો:



આકૃતિ 16. Palindrome Check

## મેમરી ટ્રીક

"સાફ કરો, ઉલટાવો, સરખાવો"

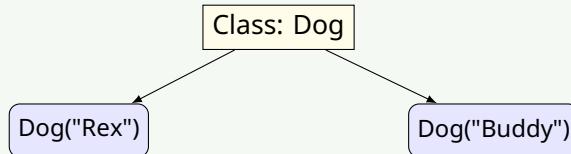
## પ્રશ્ન 5(a) [3 ગુણ]

કલાસ અને ઓફ્જેક્ટને વ્યાખ્યાયિત કરો અને ઉદાહરણ સાથે સમજાવો.

## જવાબ

- કલાસ: બ્લુપ્રિન્ટ/ટેમ્પલેટ.
- ઓફ્જેક્ટ: કલાસનો ઇન્સ્ટન્સ.

## સંબંધ ડાયાગ્રામ:



આકૃતિ 17. Instantiation

મેમરી ટ્રીક

``CAMBO''

## પ્રશ્ન 5(b) [4 ગુણ]

કન્સ્ટ્રક્ટરનું વર્ગીકરણ કરો. જેમાંથી એકને વિગતે સમજાવો.

જવાબ

પ્રકારો: Default, Parameterized, Non-parameterized, Copy.

**Parameterized Constructor:**

```

1 class Student:
2     def __init__(self, name):
3         self.name = name
4 s = Student("Alice")
  
```

એક્ઝિક્યુશન ફ્લો:



આકૃતિ 18. Constructor Lifecycle

મેમરી ટ્રીક

``PICAN''

## પ્રશ્ન 5(c) [7 ગુણ]

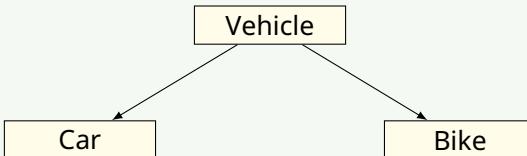
hierarchical inheritance માટે Python કોડ વિકસાવો અને સમજાવો.

જવાબ

```

1 class Vehicle:
2     def start(self): print("Engine On")
3
4 class Car(Vehicle):
5     def drive(self): print("Driving")
6
7 class Bike(Vehicle):
8     def ride(self): print("Riding")
9
10 c = Car(); c.start()
11 b = Bike(); b.start()
  
```

## ઇનહેરિટન્સ ટ્રી:



આકૃતિ 19. Hierarchical Inheritance

## મેમરી ટ્રીક

“પેરેન્ટ્સ શેર કરે, ચિલ્ડ્રન સ્પેશિયલાઇઝ કરે”

## પ્રશ્ન 5(a OR) [3 ગુણ]

Python માં `__init__` method શું છે? તેના હેતુને યોગ્ય ઉદાહરણ સાથે સમજાવો.

## જવાબ

ખાસ મેથડ જે ઓફ્જેક્ટ બનાવતી વખતે એટ્રિબ્યુટ્સ શરૂ કરવા માટે આપોઆપ કોલ થાય છે.

ઉદાહરણ:

```

1 class Rect:
2     def __init__(self, w, h):
3         self.w = w
4         self.h = h
  
```

## મેમરી ટ્રીક

“ASAP: Attributes Set At Production”

## પ્રશ્ન 5(b OR) [4 ગુણ]

Python class માટે methods નું વર્ગીકરણ કરો. તે માંથી એકને વિગતવાર સમજાવો.

## જવાબ

કોષ્ટક 11. મેથડ પ્રકારો

પ્રકાર	એક્સેસ	ડેકોરેટર
Instance	self	None
Class	cls	@classmethod
Static	None	@staticmethod

## Instance Method:

```

1 def display(self):
2     print(self.name)
  
```

## મેમરી ટ્રીક

“SIAM: Self Is Always Mentioned”

## પ્રશ્ન 5(c OR) [7 ગુણ]

પોલીમોર્ફિઝમ માટે Python કોડ વિકસાવો અને સમજાવો.

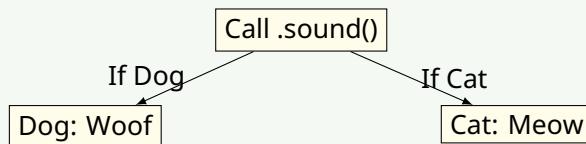
## જવાબ

```

1 class Dog:
2     def sound(self): return "Woof"
3
4 class Cat:
5     def sound(self): return "Meow"
6
7 animals = [Dog(), Cat()]
8 for a in animals:
9     print(a.sound())

```

પોલીમોર્ફિઝમ ડાયગ્રામ:



આકૃતિ 20. Dynamic Binding

## મેમરી ટ્રીક

“એક મેથડ, વિવિધ વર્તન”