ⓘ To avoid any leaks of catalog questions, this report must not be shared with candidates.

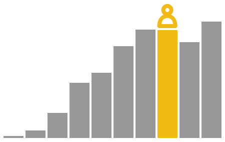
Pramitha (2018csc027@univ.jfn.ac.lk)

Campaign: Python Developer - Junior **Domain(s):** Python 3 **Language:** English **Date:** Jan 4, 2023

BETTER THAN

64%

of professionals



RANK	1 / 1
DURATION	0h31 / 0h55
SCORE	772 / 990 (78%)

Python 3 772 / 990pts (78%)

BETTER THAN **64%** of professionals

Language knowledge	🟩🟩🟩🟩🟩🟩🟩🟩🟩🟩	240 / 240pts
Problem solving	🟩🟩🟩🟩🟩🟩🟩🟩⬜⬜	408 / 558pts
Reliability	🟩🟩🟩🟩🟩🟩🟩⬜⬜⬜	124 / 192pts

Access detailed report

Question 1: Object instantiation

Python 3 00:45 / 00:45 20 / 20 pts

The candidate ran out of time for this question. Their answer was automatically submitted at the end of the preset time.

Question

How does one create a new instance `point` of the following object:

```
class Point():
    def __init__(self, x, y):
        self.x = x
        self.y = y
    def __eq__(self, other):
        return (self.x, self.y) == (other.x, other.y)
```

Answer

○ `point = new Point(x, y)`

○ `point = Point(point, x, y)`

● **`point = Point(x, y)`**

Result

✓ Correct answer
Language knowledge +20pts

Question 2: Execution order

Python 3 01:00 / 01:00 40 / 40 pts

The candidate ran out of time for this question. Their answer was automatically submitted at the end of the preset time.

Question

The code below is in a file called `file.py`. If you run the `python3 file.py` command, in what order will the code blocks be executed?

```python
#code block A - start
# ...
#code block A - end

def main():
    #code block B - start
    # ...
    #code block B - end

if __name__ == '__main__':
    main()

#code block C - start
# ...
#code block C - end
```

Answer

● A then B then C

○ only B is executed

○ A then B

○ A then C then B

○ A then C

Result

✓ Correct answer
Language knowledge +40pts

Question 3: append()

Python 3 00:30 / 00:30 40 / 40 pts

The candidate ran out of time for this question. Their answer was automatically submitted at the end of the preset time.

Question

Which of these instructions adds `5` to the following list?
`arr = [1,2,3,4]`

Answer

○ `arr.add(5)`

● `arr.append(5)`

○ `arr.push(5)`

○ `arr += 5`

Result

✓ Correct answer
Language knowledge +40pts

Question 4: Existence of key in a dict

Python 3 01:00 / 01:00 40 / 40 pts

The candidate ran out of time for this question. Their answer was automatically submitted at the end of the preset time.

Question

Which of these instructions can you use to check if the key `"Bob"` is present in the `phonebook` dictionary?

Answer

◉ **`"Bob" in phonebook`**

○ `phonebook["Bob"] is not None`

○ `phonebook["Bob"] != None`

○ `phonebook.Bob != None`

○ `phonebook.contains("Bob")`

Result

✓ Correct answer
Language knowledge +40pts

Question 5: Correction

Python 3 01:32 / 05:00 1x (3 sec) 100 / 100 pts

Question

The following `factorial` function written by your colleague Fred is supposed to return the factorial of a number, but it is broken.

Fix the `factorial` function.

As a reminder: `factorial(n) = 1 * 2 * 3 * ... * n`

Answer

```python
# Python code below
# Use print("messages...") to debug your solution.

def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
```

▶ Watch code playback

Result

✓ The factorial function works again
Language knowledge +100pts

Question 6: Largest wins from chaos

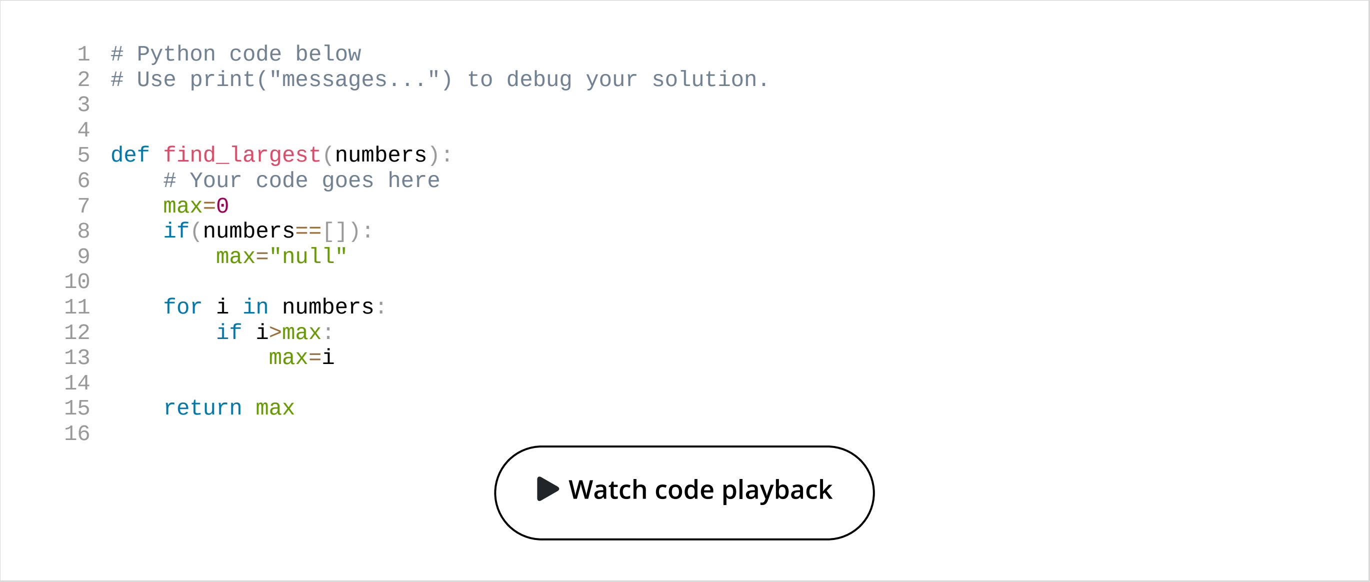
Python 3   04:04 / 05:00   1x (12 sec)   32 / 100 pts

Question

`find_largest(numbers)` should return the largest number from `numbers`. The array `numbers` always contains at least one number.

Implement `find_largest(numbers)`.

Answer



```python
# Python code below
# Use print("messages...") to debug your solution.


def find_largest(numbers):
    # Your code goes here
    max=0
    if(numbers==[]):
        max="null"

    for i in numbers:
        if i>max:
            max=i

    return max

```

Result

✓ It works using simple data sample
Problem solving +32pts

- ✗ Still works when the array contains only Integer.MIN_VALUE
  Reliability ~~+58pts~~

- ✗ Still works if the largest number is at position 0 in the array
  Reliability ~~+5pts~~

- ✗ Still works if the largest number is at the last position in the array
  Reliability ~~+5pts~~

Question 7: Robotic packages classifier

Python 3 15:00 / 15:00 10x (2 min) 0 / 150 pts

The candidate ran out of time for this question. Their answer was automatically submitted at the end of the preset time.

Question

Objective
Sort the packages using the robotic arm of the factory.

Rules
You work in an automated factory and your objective is to write the function that will dispatch the packages to the correct stack, according to their volume and mass. A package is bulky if its volume (Width x Height x Length) is greater than or equal to 1,000,000 cm^3 or when one of its dimension is greater or equal than 150 cm. A package is heavy when its mass is greater or equal than 20 kg. You must dispatch the packages in the following stacks: STANDARD : standard packages (those which are not bulky nor heavy) can be handled normally. SPECIAL : packages that are either heavy or bulky can't be handled automatically. REJECTED : packages that are both heavy and bulky are rejected.

Implementation
Implement the function `solve(width, height, length, mass)` (units are centimeter for the dimensions and kilogram for the mass). This function must return a string: the name of the stack where the package should go.

Victory Conditions
All the packages are on the proper stack.

Lose Conditions
Your function returns a wrong answer.
Constraints
20 ≤ width, height, length ≤ 200
10 ≤ mass ≤ 1000

Answer

```python
import sys
import math
from contextlib import redirect_stdout


def solve(width, height, length, mass):
    # Write your code here
    v=width*height*length
    d=mass/v

    if(v>=1000000 and d>=150 and mass>=20 ):
        return "rejected"
    elif(v>=1000000 and d>=150):
        return "can handle"
    elif(mass>=20):
        return "can handle"

    else:
        return 'can be handled normaly '
    # To debug: print("Debug messages...", file=sys.stderr, flush=True)



# Ignore and do not change the code below
def main():
    # pylint: disable = C, W

    # game loop
    while True:
        width, height, length, mass = [int(i) for i in input().split()]
        with redirect_stdout(sys.stderr):
            if(width,height,le<=200 and width >=20)
            action = solve(width, height, length, mass)
        print(action)


if __name__ == "__main__":
    main()
# Ignore and do not change the code above
```

▶ Watch code playback

Result

⊗ Many boxes
Problem solving ~~+40pts~~

⊗ Only STANDARD and REJECTED boxes
Problem solving ~~+40pts~~

⊗ Only STANDARD and SPECIAL boxes
Problem solving ~~+35pts~~

⊗ Test limits
Problem solving ~~+35pts~~

Question 8: Approximation of π

Python 3 05:16 / 12:00 3x (1 min) 200 / 200 pts

Question

In this exercise we will calculate an approximation of π (Pi).

The technique is as follows:
Take a random point P at coordinate (x, y) such that `0 # x # 1` and `0 # y # 1`. If `x^2 + y^2 # 1`, then the point is inside the quarter disk of radius `1`, otherwise the point is outside.

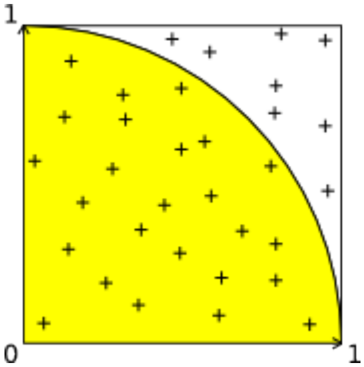


Fig 1. An example using 33 random points.

We know that the probability that the point is inside the quarter disk is equal to `#/4`.

Write the `piApprox(pts)` function who will use the points `pts` to return an approximation of the number float π.
`pts` is a multidimentional list of float.

Input:
Each item in `pts` is a point. A point is represented by an array containing exactly two numbers, respectively, `x` and `y` such that `0 # x # 1` and `0 # y # 1`. `pts` is never None and always contains at least one item.

Answer

```python
# Python code below
# Use print("messages...") to debug your solution.

def pi_approx(pts):
    # Your code goes here
    SatisfyTheCircle=[]
    for i in range(0,len(pts)):
        point=pts[i]
        x=point[0]
        y=point[1]

        result=x**2 + y**2
        if(result<=1):
            if(x==1 and y==0):
                print("point(1,0) is inside")
            SatisfyTheCircle.append(point)

    #print("point(1,0) is here: ", val==1)
    aproximateValue=4*(len(SatisfyTheCircle)/len(pts))
    return aproximateValue

```

▶ Watch code playback

Result

✓ Approximation of π is correct (related to pts)
Problem solving +171pts

✓ The point P(1, 0) is inside the quarter disk
Reliability +29pts

Question 9: Summer Sales

Python 3 01:54 / 15:00 2x (26 sec) 300 / 300 pts

Question

It's almost the Summer Sales!

You work for a shop that wishes to give a discount of `discount%` to the most expensive item purchased by a given customer during the sales period. Only one product can benefit from the discount.

You are tasked by the shop owner to implement the function `calculate_total_price(prices, discount)` which takes the list of prices of the products purchased by a customer and the percentage `discount` as parameters and returns the total purchase price as an integer (rounded down if the total is a float number).

Constraints:
0 ≤ `discount` ≤ 100 0 < price of a product < 100000 0 < number of products < 100

Answer

```python
import sys
import math
from contextlib import redirect_stdout


def calculate_total_price(prices, discount):
    # Write your code here
    x=max(prices)
    if x>=100000:
        print("this price has large discount ")
    prices.remove(x)
    value=x-x*(discount/100)
    prices.append(value)
    print(prices)
    sum=0
    for i in prices:
        sum+=i
    return int(sum)



    # To debug: print("Debug messages...", file=sys.stderr, flush=True)



# Ignore and do not change the code below
def main():
    # pylint: disable = C, W
    discount = int(input())
    n = int(input())
    prices = [int(i) for i in input().split()]
    with redirect_stdout(sys.stderr):
        price = calculate_total_price(prices, discount)
    print(price)


if __name__ == "__main__":
    main()
# Ignore and do not change the code above
```

▶ Watch code playback

Result

✓ Simple sum
Problem solving +35pts

✓ Good sale
Problem solving +35pts

✓ Large discount
Problem solving +35pts

✓ Correct rouding
Reliability +35pts

✓ One item free
Problem solving +35pts

✓ No sales
Problem solving +35pts

✓ Big purchase
Problem solving +30pts

✓ Same price
Reliability +30pts

✓ One item only
Reliability +30pts

Glossary

Language knowledge

Measuring this skill allows us to determine the candidate's level of experience in the practice of a specific programming language. **This skill is particularly important if, for example, you are looking for a developer who wil have to become quickly operational.**

Design

This measurement gives an indication of the candidate's ability to implement standard solutions to common problems. A developer with a good level of proficiency in this skill will contribute to increase the quality (maintainability, extensibility) of your applications. It does not rely specifically on technology. **This skill is particularly important if, for example, you are looking for a developer who will have to work on the architecture of your applications and to develop long-term solutions.**

Problem solving

This skill corresponds to the candidate's ability to understand and to structure their reasoning in order to find efficient solutions to complex problems. It does not rely specifically on technology. **This skill is particularly important if, for example, you are looking for R&D developers.**

Reliability

Reliability refers to the candidate's ability to achieve solutions that address specific cases. Developers with a high reliability score are likely to create more robust applications (less bugs).