

問題1

以下のように出力されるプログラムを選択しなさい

1人につき3つまで購入可能
ケンが購入しました

①	②
<pre>class User: def __init__(self, name): self.name def buy(self, num): if num > 3: print("1人につき3つまで購入可能") else: print(self.name + "が購入しました") ken = User("ケン") ken.buy(5) ken.buy(3)</pre>	<pre>class User: def __init__(self, name): self.name = name def buy(self, num): if num > 3: print("1人につき3つまで購入可能") else: print(self.name + "が購入しました") ken = User("ケン") ken.buy(5) ken.buy(3)</pre>

問題2

以下のように出力されるプログラムを選択しなさい

残り4人です
20分待ちです
15分待ちです
10分待ちです
5分待ちです
あなたの番です

①	②
<pre>import asyncio import time async def show_waiting_time(num): while num > 0: waiting_time = num * 5 print(str(waiting_time) + "分待ちです") num = num - 1 time.sleep(1) print("あなたの番です") people_num = 4 show_waiting_time(people_num) print("残り" + str(people_num) + "人です")</pre>	<pre>import asyncio import time async def show_waiting_time(num): while num > 0: waiting_time = num * 5 print(str(waiting_time) + "分待ちです") num = num - 1 time.sleep(1) print("あなたの番です") people_num = 4 asyncio.create_task(show_waiting_time(people_num)) print("残り" + str(people_num) + "人です")</pre>

実装式問題

コップの容量capacityと、現在コップに入っている水量volumeが与えられる
コップにあと何ml水を入れることができるかを計算して出力するプログラムを完成させなさい

- ・ capacityは1以上1000以下の整数とする
- ・ volumeは0以上1000以下の整数とする
- ・ volumeの値はcapacityの値以下とする

例) capacity=100, volume=70のとき
あと30ml入ります

```
class Glass:
    def __init__(self, cap, vol):
        self.capacity = cap
        self.volume = vol

#===回答ここから===

#===回答ここまで===

capacity = 100 # 可変
volume = 70 # 可変
glass = Glass(capacity, volume)

print("あと" + str(glass.remaining_capacity()) + "ml入ります")
```

以上で問題は終了です。次ページより解答となります。

解答

問題1

正解選択肢

②

```
class User:
    def __init__(self, name):
        self.name = name

    def buy(self, num):
        if num > 3:
            print("1人につき3つまで購入可能")
        else:
            print(self.name + "が購入しました")

ken = User("ケン")

ken.buy(5)
ken.buy(3)
```

問題2

正解選択肢

②

```
import asyncio
import time

async def show_waiting_time(num):
    while num > 0:
        waiting_time = num * 5
        print(str(waiting_time) + "分待ちです")
        num = num - 1
        time.sleep(1)

    print("あなたの番です")

people_num = 4

asyncio.create_task(show_waiting_time(people_num))
print("残り" + str(people_num) + "人です")
```

実装式問題

解答例

```
def remaining_capacity(self):
    return self.capacity - self.volume
```