

```

gateauChoc = {'oeufs':4,'sucre':150,'farine':80,'beurre':200,'chocolat':200}
quatreQuarts = {'oeufs':4,'sucre':250,'farine':250,'beurre':250}
lesRecettes = {'omelette':{ 'oeufs': 4, 'lait (en cl)': 5},'soupe' : { 'poireau': 4, 'pommes de
terre':2},'fondant au chocolat' : gateauChoc,'quatre-quarts' : quatreQuarts}

```

exercice 1

```

def recettePossible(frigo, recette):
    for k, v in frigo.items():
        if k not in recette:
            return False
        return True

```

exercice 2

```

def ajouteCourses(frigo, courses):
    for k, v in courses.items():
        if k not in frigo:
            frigo[k] = v
        else:
            frigo[k] += v
    return frigo

```

exercice 3

```

def recettePossible(frigo,courses):
    """
    >>> recettePossible({'oeufs':4, 'sucre':350, 'farine':300,'chocolat':200, 'beurre':250}, lesRecettes)
    ['fondant au chocolat','quatre-quarts']
    """
    recette = []
    for k in courses.values():
        if recettePossible(frigo, k) == True:
            recette.append(k)
    return recette

```

exercice 4

```

def cuisineRecette(frigo, uneRecette):
    """
    >>> cuisineRecette({'oeufs':4, 'sucre':350, 'farine':300,'chocolat':200, 'beurre':250}, gateauChoc)
    {'sucre': 200, 'farine': 220, 'beurre': 50}
    """
    for k, v in uneRecette.items():
        if k in frigo:
            frigo[k] -= v
        if frigo[k] == 0:
            del frigo[k]
    return frigo

```

exercice 5

```
def cuisineLesRecettes(frigo, LesRecettes):
    for k, v in LesRecettes.items():
        frigo -= cuisineLesRecettes(frigo, k)
    return frigo
```

exercice 6

```
def CoursePourRecette(frigo, uneRecette):
    listecourse = {}
    for k, v in uneRecette.items():
        if k not in frigo:
            listecourse[k] = v
    return listecourse
```

exercice 7

```
def ToutesLesCourses(frigo, LesRecettes):
    listeCourses = {}
    for k, v in LesRecettes.items():
        listeCourses += CoursePourRecette(frigo, k)
    return listeCourses
```

exercice 8

```
def lesIngredients(recette, lesRecettes):
    return ToutesLesCourses("", recette)
```

exercice 9

```
laClassification = {'recettes végétariennes' : ['omelette', 'soupe'], 'entrées' : ['soupe'], 'desserts' :
['quatre-quarts', 'fondant au chocolat'] }
```

```
def categoriesDUneRecette(recette, laClassification):
    """
    >>> categoriesDUneRecette('soupe', laClassification)
    ['recettes végétariennes', 'entrées']
    """
    l = []
    for k, v in laClassification.items():
        if recette in v:
            l.append(k)
    return l
```

exercice 10

```
def RecettesDUneCategorie(categ, laClassification, recettes):
    """
    >>> RecettesDUneCategorie('recettes végétariennes', laClassification, ['escargots de Bourgogne',
'soupe', 'carbonnade', 'tarte au sucre'])
```

```

['soupe']
"""
l = []
for elt in recettes:
    if elt in laClassification[categ]:
        l.append(elt)
return l

```

exercice 11

```

def convient(repas , categ , laClassification):
    """
    >>> convient(['soupe','carbonnade','tarte au sucre'], ['recettes
végétariennes','poissons'],laClassification)
    False
    >>> convient(['escargots de Bourgogne','soupe','fondant au chocolat'], ['recettes
végétariennes','entrées'],laClassification)
    True
    """
    for elt in categ:
        if elt not in laClassification:
            return False
        elif repas in laClassification[elt]:
            del categ[elt]
    if categ == {}:
        return True
    return False

```

```

if __name__ == "__main__":
    import doctest
    doctest.testmod(verbose = True)

```