Exercice 1 :

```python
class Pile:
    def __init__(self):
        self.donnees = []

    def estVide(self):
        return self.donnees == []

    def empiler(self, e):
        self.donnees.append(e)

    def depiler(self):
        if not self.estVide():
            return self.donnees.pop()

    def __str__(self):
        affiche =""
        for x in self.donnees:
            affiche = str(x)+"\n"+affiche
        return affiche

class File:
    def __init__(self):
        self.donnees = []

    def estVide(self):
        return self.donnees == []

    def enfiler(self, e):
        self.donnees.append(e)

    def defiler(self):
        if not self.donnees == []:
            return self.donnees.pop(0)

    def __str__(self):
        affiche = ""
        for x in self.donnees:
            affiche += str(x) + " "
        return affiche
```

Exercice 2 :

```python
historique = Historique()
fichier_histo = open("historique.csv")

for lien in csv.DictReader(fichier_histo, delimiter=";"):
    # le noeud sera un dictionnaire
    historique.empiler(lien)

print(historique.retour())
historique.nouvelle_adresse("https://docs.python.org/fr/3/")
```

```python
print(historique.retour())

fichier_histo.close()
```

```python
    def retour(self)->str:
        res = self.depiler()
        if res is not None:
            return res["site"]
        else:
            return "Pas d'historique"

    def nouvelle_adresse(self, lien: str)->None:
        self.empiler({"date":datetime.now().strftime("%Y-%m-%d %H
            :%M"),
                      "site": lien})
```

Exercice 3 :

```python
class File:
    def __init__(self):
        self.gauche = Pile()
        self.droite = Pile()

    def enfiler(self, e: int)->None:
        self.gauche.empiler(e)

    def defiler(self)->int:
        if self.droite.est_vide():
            while not self.gauche.est_vide():
                self.droite.empiler(self.gauche.depiler())
        return self.droite.depiler()
```

Exercice 4 :

```python
from mod_file import File

#Création du cercle
soldats = File()

for i in range(1,42):
    soldats.enfiler(i)

#Élimination tous les 3
while not(soldats.est_vide()):
        for _ in range(2):
                soldats.enfiler(soldats.defiler())

    #soldat éliminé
        elimine = soldats.defiler()

#dernier éliminé
print(elimine)
```

Exercice 5 :

```python
from mod_pile import Pile

def bien_parenthesee(code: str)->bool:
    parentheses = Pile()
    taille = len(code)
    i = 0
    correct = True
    while i < taille and correct:
        if code[i] == "(":
            parentheses.empiler("(")
        elif code[i] == ")":
            if parentheses.depiler() is None:
                correct = False
        i += 1

    return correct

chaine_correcte = "((e)ee(e))"
chaine_incorrecte = "(e))"

print(bien_parenthesee(chaine_correcte))
print(bien_parenthesee(chaine_incorrecte))
```

Exercice 6 :

```python
from mod_pile import Pile

def polonaise(chaine: str)->int:
    p = Pile()
    for e in chaine.split():
        if e == "+" or e == "*":
            val1 = p.depiler()
            val2 = p.depiler()
            if e == "+":
                p.empiler(val1+val2)
            else:
                p.empiler(val1*val2)
        else:
            p.empiler(int(e))

    return p.depiler()

chaine = "1 2 3 * + 4 *"
print(polonaise(chaine))
```