TP 4: Arithmétique

DRAME Mariam et BLEUSE Chloé

December 7, 2017

1 Un nombre est-il premier?

1.1 Quotient et reste en python

Pour obtenir en python le quotient d'une division euclidienne de a par b, il faut effectuer la commande suivante: print(a/b). De même pour le reste la commande est : print(a%b)

1.2 Définition d'un nombre premier

Un nombre premier est un entier naturel qui admet exactement deux diviseurs distincts entiers et positifs qui sont 1 et lui-même.

1.3 Exemples de nombres premiers

1001 n'est pas un nombre premier car il est au moins divisible par 7. 2017 est un nombre premier, tout comme 3001,49999. 89999 n'est pas un nombre premier car il est divisible par 7.

1.4 Programme en python pour déterminer si un nombre est premier ou non

Le programme qui détermine si un nombre est premier ou non est le suivant:

```
def is_prime(n):
        i=1
        for k in range (2,n):
                i=n%k
                if i==0:
                        return ('false')
        return ('true')
```

Dans cette fonction, il faut remarquer que le résultat est un booléen et non une chaîne de caractère. Cette précision sera importante pour la suite du TP.

1.5 Nombres de Fermat

Pour déterminer si les nombres des Fermat F_n pour n=0,1,2,3,4,5, nous les avons testé avec le programme précédent. Nous avons ainsi déterminé que F_0, F_1, F_3 et F_4 sont des nombres premiers tandis que F_2 ne l'ai pas. Cependant pour F_5, le nombre est trop grand donc le programme renvoie un message d'erreur qui indique que la mémoire est insuffisante.

2 Crible d'Eratosthène, distribution des nombres premiers

2.1 Liste des nombres premiers inférieurs à 200

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130
131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170
171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190
191	192	193	194	195	196	197	198	199	200

Pour trouver les nombres premiers inférieurs à 200, on a construit un crible d'Eratosthène allant jusqu'à 200. Puis, on a surligner les multiples de 2 en jaune, ceux de 3 en rose, ceux de 5 en bleu et ceux de 7 en vert. Les nombres restants (ceux écrit en rouge) sont donc les nombres premiers inférieurs à 200.

2.2 Programme en python pour déterminer les nombres premiers $\leq n$

Le programme qui renvoie la liste de tous les premiers inférieurs à n :

```
def primes(n):
        L = []
        for i in range(2, n+1):
                if is_prime(i):
                        L.append(i)
        return L
```

2.3 Exemple pour n=1000

Voici la liste de tout les premiers inferieurs à 1000:

```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29,
31, 37, 41, 43, 47, 53, 59, 61, 67, 71,
73, 79, 83, 89, 97, 101, 103, 107, 109, 113,
127, 131, 137, 139, 149, 151, 157, 163, 167, 173,
179, 181, 191, 193, 197, 199, 211, 223, 227, 229,
233, 239, 241, 251, 257, 263, 269, 271, 277, 281,
283, 293, 307, 311, 313, 317, 331, 337, 347, 349,
353, 359, 367, 373, 379, 383, 389, 397, 401, 409,
419, 421, 431, 433, 439, 443, 449, 457, 461, 463,
467, 479, 487, 491, 499, 503, 509, 521, 523, 541,
547, 557, 563, 569, 571, 577, 587, 593, 599, 601,
607, 613, 617, 619, 631, 641, 643, 647, 653, 659,
661, 673, 677, 683, 691, 701, 709, 719, 727, 733,
739, 743, 751, 757, 761, 769, 773, 787, 797, 809,
811, 821, 823, 827, 829, 839, 853, 857, 859, 863,
877, 881, 883, 887, 907, 911, 919, 929, 937, 941,
947, 953, 967, 971, 977, 983, 991, 997]
```

Pour la determiner, il a simplement fallu appliquer à 1000 la fonction python $primes$.

3 Factorisation d'un entier en premiers

3.1 Théorème fondamental de l'arithmétique

Tout entier strictement positif peut être écrit comme un produit de nombres premiers d'une unique façon, à l'ordre près des facteurs.

3.2 Décomposition en facteurs premiers de 924

Les étapes de la décomposition de 924 on a :

```
924/2=462 ( 2 est le plus petit nombre qui divise le nombre 924)
462/2=231 ( 2 est le plus petit nombre qui divise le nombre 462)
231/1=77  ( 3 est le plus petit nombre qui divise le nombre 231)
77/7=11   ( 7 est le plus petit nombre qui divise le nombre 77 )
11/11=1   ( 11 est un nombre premier la décomposition est donc terminée)

alors la décompostion en facteurs premiers: 924 = 2*2*3*7*11
```

3.3 Programme en python pour déétерminer la décomposition en facteurs premiers

4 PGCD de deux entiers, identité de Bezout, algorithme d'Euclide

4.1 Définition du PGCD

Le plus grand commun diviseur ou PGCD de deux nombres entiers non nuls est le plus grand entier qui les divise simultanément.

4.2 PGCD de 4864 et 3458

$4864 = 2^8 \times 19$

$3458 = 2 \times 1729 = 2 \times 7 \times 247 = 2 \times 7 \times 13 \times 19$

Donc: $PGCD(4854, 3458) = 19$

4.3 Identité de Bézout

L'identité de Bézout s'énonce par $ax + by = PGCD(a, b)$ où x et y sont des inconnues entières relatives et a et b sont des coefficients entiers relatifs.