プログラミング課題2020

- 計算機数学I・II
- 数値計算法I・II
- データ構造とアルゴリズム
- 数理モデル

先輩が選んだ簡単問題より

(1) Hello World (ITP1_1_A)

```
print("Hello World")
```

(2) xの3乗 (ITP1_1_B)

```
x = int(input())

print(x ** 3)
```

- `x ** y` ... x^y

(3) ワード (Volume3_0335)

```
W = int(input())

print(W * 32)
```

(4) 華氏と摂氏 (Volume3_0380)

```
F = int(input())

print((F - 30) // 2)
```

- `x // y` ... $x ÷ y$ の整数部分(小数点切り捨て)

(5) 参加者数 (Volume3_0315)

```
p, m, c = map(int, input().split())
```

```
print(p + m + c)
```

- `map(関数, list)` ... イテラブルの全要素に関数を適用する
- イテラブル ... 繰り返し可能なオブジェクト
    - リストやタプルやrange関数で作成したオブジェクトのこと
- `split(区切り文字, 最大分割数)` ... 区切り文字で分割する
    - 区切り文字
        - `","` ... 単独文字で区切る
        - `"[-+#]"` ... `[]`で囲まれた中の任意の1文字で分割する
        - `"XXX|YYY|ZZZ"` ... `|`で区切ったいずれかのパターンで分割する。複数文字も指定可能

(6) 長方形の面積と周の長さ (ITP1_1_C)

```
a, b = map(int, input().split())

s = a * b
l = a * 2 + b * 2

print(s, l)
```

(7) 大小関係 (ITP1_2_A)

```
a, b = map(int, input().split())

if a == b :
    print("a == b")
elif a < b :
# a == b ではなく a < b のとき
    print("a < b")
else :
# 上記条件のどれにも当てはまらないとき
    print("a > b")
```

(8) 範囲 (ITP1_2_B)

```
a, b, c = map(int, input().split())

if a < b and b < c :
# 条件式は and, or を使うこともできる
    print("Yes")
else :
    print("No")
```

(9) お年玉 (Volume3_0357)

```
a, b = map(int, input().split())

print((a + b) // 2)
```

(10) 赤とんぼ (Volume3_0381)

```
x1, x2 = map(int, input().split())

print(abs(x1 - x2))
```

- `abs()` ... `()`内の絶対値を求める

入力と出力をマスターしよう

(11) 乗車券 (Volume2_0257)

```
B = input()

O = ["1 1 0", "0 0 1"]

if B in O :
    print("Open")
else :
    print("Close")
```

- `x in y` ... `x`が`y`に含まれているとき`True`を返す

(12) 時計(ITP1_1_D)

```
S = int(input())

h = S // 3600
m = (S - h * 3600) // 60
s = S - (h * 3600 + m * 60)

print(h, ":", m, ":", s, sep = "")
```

- `sep` ... `print文`の区切り文字を指定
    - デフォルトは`" "`

(13) A/B Problem(割り算) (ITP1_4_A)

```
a, b = map(int, input().split())

d = a // b
r = a - b * d
f = a / b

print(f"{d} {r} {f:.6f}")
```

- print(f"文字列") ... フォーマット文字列
  - 文字列中の置換フィールドに変数をそのまま指定
  - {} ... 置換フィールド
  - {x:書式指定文字列} ... xを書式指定文字列で指定した通りに表示する
    - {x:.6f} ... 有効数字指定
      - xの小数点7桁目を四捨五入し、6桁までを表示

(14) 円の面積と円周 (ITP1_4_B)

```
import math

r = float(input())

s = r * r * math.pi
l = r * 2 * math.pi

print(f"{s:.6f} {l:.6f}")
```

- math ... 数学関数を提供するモジュール
  - math.pi ... $π$

(15) 坪面積の計算 (Volume0_0094)

```
a, b = map(int, input().split())

S = a * b / 3.305785

print(f"{S:.6f}")
```

(16) 3つの数の整列 (ITP1_2_C)

```
a, b, c = map(int, input().split())

if a < b :
    if b < c :
        print(a, b, c)
    elif a < c :
```

```
        print(a, c, b)
    else :
        print(c, a, b)
elif b < a :
    if a < c :
        print(b, a, c)
    elif b < c :
        print(b, c, a)
    else :
        print(c, b, a)
```

(17) 長方形の中の円 (ITP1_2_D)

```
W, H, x, y, r = map(int, input().split())

if x - r >= 0 and y - r >= 0 :
    if x + r <= W and y + r <= H :
        print("Yes")
    else :
        print("No")
else :
    print("No")
```

(18) 長方形 (Volume3_0345)

```
e1, e2, e3, e4 = map(int, input().split())
E = [e1, e2, e3, e4]

e = set(E)
e_list = list(e)

if len(e) == 2 :
    if E.count(e_list[0]) == 2 :
        print("yes")
    else :
        print("no")
else :
    print("no")
```

- set型 ... 重複しない要素のコレクション。集合演算も可能なミュータブルな型
- ミュータブル ... 要素を追加したり削除したりできる
- set() ... ()内のリストやタプルから重複した要素を取り除く
    - 元のリストの順序は保持されない
- len() ... ()内のリストや文字列の長さを求める
- X.count(y) ... X内にあるyの数を求める

(19) ペンシル (Volume6_0641)

```
N, A, B, C, D = map(int, input().split())

a = N // A
c = N // C

if a * A < N :
    a += 1

if c * C < N :
    c += 1

b = a * B
d = c * D

if b < d :
    print(b)
else :
    print(d)
```

(20) ソーシャルゲーム (Volume6_0652)

<while文を使った書き方>

```
A, B, C = map(int, input().split())

day = 0
coin = 0

while coin < C :
    day += 1
    coin += A
    if day % 7 == 0 :
        coin += B

print(day)
```

- x % y ... $x÷y$ の余り

<数式と場合わけを使った書き方>

```
A, B, C = map(int, input().split())

week = C // (7 * A + B)
coin = week * (7 * A + B)
day = week * 7

if (C - coin) / A < 7 :
    day += (C - coin) // A
```

```
    coin += (C - coin) // A * A

    if coin != C :
        day += 1

else :
    day += 7
    coin += 7 * A + B

    if coin < C :
        day += 1

print(day)
```

繰り返し

(21) 複数のHelloWorld (ITP1_3_A)

```
for i in range(1000) :
    print(Hello World)
```

(22) チケットの売り上げ (Volume2_0277)

```
dic = {1 : 6000, 2 : 4000, 3 : 3000, 4 : 2000}

for i in range(4) :
    t, n = map(int, input().split())
    print(dic[t] * n)
```

- 辞書型 ... {}内にkeyとvalueの組み合わせが含まれているデータ
- {key:value}
- dic[] ... []内にkeyを指定することで対応するvalueを表示する

(23) 気温の差 (Volume2_0276)

```
for i in range(7) :
    a, b = map(int, input().split())
    print(a - b)
```

(24) お化け屋敷 (Volume1_0173)

```

```

```
for i in range(9) :
    e, a, b = map(str, input().split())
    a = int(a)
    b = int(b)
    print(e, a + b, a * 200 + b * 300)
```

(25) 10問解いたら何点取れる？ (Volume2_0256)

```
s = 0

for i in range(10) :
    s += int(input())

print(s)
```

(26) テストケースの出力 (ITP1_3_B)

```
i = 0

while True :
    x = int(input())
    if x == 0 :
        break
    else :
        i += 1
        print("Case ", i, ": ", x, sep = "")
```

(27) 2つの数の交換 (ITP1_3_C)

```
while True :
    x, y = map(int, input().split())
    if x == 0 and y == 0 :
        break
    else :
        if x < y :
            print(x, y)
        else :
            print(y, x)
```

(28) 直角三角形 (Volume0_0003)

```
N = int(input())
```

```
for i in range(N) :
    a, b, c = map(int, input().split())
    L = [a, b, c]

    m = L.pop(L.index(max(L)))
    s = L[0] ** 2 + L[1] ** 2

    if m ** 2 == s :
        print("YES")
    else :
        print("NO")
```

- max() ... ()内のリストや文字列で最大の値を表示
- X.index(y) ... yがXの何番目にあるかを表示
- X.pop() ... ()内で指定した位置の要素を削除し、その値を取得する
    - 引数を省略すると、末尾の要素を削除し取得する
    - 引数に負の値を入力すると、末尾からその値分の位置の要素を削除し取得する

(29) 借金 (Volume0_0007)

```
n = int(input())
s = 100000

for i in range(n) :
    s *= 1.05
    m = s % 1000
    if m != 0 :
        s = s - m + 1000

print(int(s))
```

(30) おつり (Volume5_0521)

```
while True :
    n = int(input())
    s = 0
    if n == 0 :
        break
    else :
        m = 1000 - n

        if m >= 500 :
            s += 1
            m -= 500

        while m >= 100 :
            s += 1
            m -= 100
```

```
        if m >= 50 :
            s += 1
            m -= 50

        while m >= 10 :
            s += 1
            m -= 10

        if m >= 5 :
            s += 1
            m -= 5

        s += m

    print(s)
```

(31) 約数の数 (ITP1_3_D)

```
a, b, c = map(int, input().split())
s = 0

for i in range(a, b + 1) :
    if c % i == 0 :
        s += 1

print(s)
```

繰り返しと関数

(32) 九九 (Volume0_0000)

```
for i in range(1, 10) :
    for j in range(1, 10) :
        print(i, "x", j, "=", i * j, sep = "")
```

- `range(start, stop)` … `start`と`stop`を指定することで、指定した範囲の連番を返す
  - `stop`の値は含まない
- `range(start, stop, step)` … `step`で増加分の値も指定することができる
  - デフォルトは`1`
  - 負の値を指定すれば、逆順にすることもできる

(33) 長方形の描画 (ITP1_5_A)

```
while True :
    H, W = map(int, input().split())
    if H == 0 :
        break
    else :
        for i in range(H) :
            for j in range(W) :
                print("#", sep = "", end = "")
            print()
        print()
```

- end … print文の文末を指定
    - デフォルトは"\n"

(34) フレームの描画 (ITP1_5_B)

```
while True :
    H, W = map(int, input().split())
    if H == 0 :
        break
    else :
        for i in range(H) :
            for j in range(W) :
                if i == 0 or i == H - 1 or j == 0 or j == W - 1 :
                    print("#", sep = "", end = "")
                else :
                    print(".", sep = "", end = "")
            print()
        print()
```

(35) チェスボードの描画 (ITP1_5_C)

```
while True :
    H, W = map(int, input().split())
    if H == 0 :
        break
    else :
        for i in range(H) :
            for j in range(W) :
                if (i + j) % 2 == 0 :
                    print("#", sep = "", end = "")
                else :
                    print(".", sep = "", end = "")
            print()
        print()
```

(36) 簡単な計算機 (ITP1_4_C)

```
while True :
    a, op, b = map(str, input().split())
    a = int(a)
    b = int(b)

    if op == "?" :
        break
    else :
        if op == "+" :
            print(a + b)
        elif op == "-" :
            print(a - b)
        elif op == "*" :
            print(a * b)
        else :
            print(a // b)
```

(37) 最小値、最大値、合計値 (ITP1_4_D)

```
n = int(input())
A = [int(a) for a in input().split()]
# 空白で区切った入力文字をint型にしてAというリストの要素にする

print(min(A), max(A), sum(A))
```

- `min()` … `()`内のリストや文字列で最小の値を表示
- `sum()` … `()`内のリストの合計値を表示

(38) ICPC 得点集計ソフトウェア (Volume11_1147)

```
while True :
    n = int(input())
    P = []

    if n == 0 :
        break
    else :
        for i in range(n) :
            P.append(int(input()))

        Ps = sorted(P)
        del Ps[0]
        del Ps[n - 2]
        s = sum(Ps)

        print(s // (n - 2))
```

- `X.append(y)` ... Xの末尾にyを加える
- `sorted(X)` ... Xを昇順に並び替えた新たなリストを生成する
    - 元のリストXはそのまま保持される
- `del` ... インデックスを指定して要素を削除する

(39) 階乗の0個数 (Volume0_0052)

```
while True :
    n = int(input())

    if n == 0 :
        break
    else :
        for i in range(1, n) :
            n *= i

        N = str(n)
        l = len(N)

        s = 0
        for j in range(l - 1, -1, -1) :
            if N[j] == "0" :
                s += 1
            else :
                break

        print(s)
```

(40) フィボナッチ数列 (ALDS1_10_A)

```
n = int(input())
F = [0] * 45

def fib(n) :
    if n == 0 or n == 1 :
        return 1
    else :
        if F[n] != 0 :
            return F[n]
        else :
            F[n] = fib(n - 1) + fib(n - 2)
            return F[n]

print(fib(n))
```

- `def 関数名(引数)` ... 関数を定義する
    - `return` ... その関数を実行した時の戻り値
- DP

(41) コッホ曲線 (ALDS1_5_C)

```
import math

n = int(input())

p1 = [0.0, 0.0]
p2 = [100.0, 0.0]

def rad(r) :
    return math.radians(r)

def sin(s) :
    return math.sin(s)

def cos(s) :
    return math.cos(s)

def Koch(n, p1, p2) :
    if n == 0 :
        return 0
        # 戻り値なし
    else :
        s = [0, 0]
        t = [0, 0]
        u = [0, 0]

        s[0] = (p2[0] - p1[0]) / 3 + p1[0]
        s[1] = (p2[1] - p1[1]) / 3 + p1[1]
        t[0] = (p2[0] - p1[0]) / 3 * 2 + p1[0]
        t[1] = (p2[1] - p1[1]) / 3 * 2 + p1[1]

        u[0] = (t[0] - s[0]) * cos(rad(60)) - (t[1] - s[1]) * sin(rad(60))
+ s[0]
        u[1] = (t[0] - s[0]) * sin(rad(60)) + (t[1] - s[1]) * cos(rad(60))
+ s[1]

        Koch(n - 1, p1, s)
        pri(s[0], s[1])

        Koch(n - 1, s, u)
        pri(u[0], u[1])

        Koch(n - 1, u, t)
        pri(t[0], t[1])

        Koch(n - 1, t, p2)

def pri(x, y) :
    print(f"{x:.8f} {y:.8f}")

print(p1[0], p1[1])
```

```
Koch(n, p1, p2)
print(p2[0], p2[1])
```

- `math.radians(x)` ... 角度（degree）をラジアンに変換する
- `math.sin(x)` ... $sin(x)$ xはラジアン
- `math.cos(x)` ... $cos(x)$ xはラジアン

リスト

(42) 数列の反転 (ITP1_6_A)

```
n = int(input())
A = [int(a) for a in input().split()]

for i in range(n - 1, -1, -1) :
    if i == 0 :
        print(A[i])
    else :
        print(A[i], " ", sep = "", end = "")
```

(43) 平均点 (Volume5_0592)

```
s = 0

for i in range(5) :
    p = int(input())

    if p < 40 :
        s += 40
    else :
        s += p

print(s // 5)
```

(44) 科目選択 (Volume6_0619)

```
Si = []
So = []

for i in range(4) :
    p = int(input())
    Si.append(p)

for j in range(2) :
    p = int(input())
```

```
    So.append(p)

s = sum(Si) - min(Si) + sum(So) - min(So)

print(s)
```

(45) コンテスト (Volume5_0533)

```
W = []
K = []

for i in range(10) :
    W.append(int(input()))

for j in range(10) :
    K.append(int(input()))

Ws = sorted(W, reverse = True)
# reverse = True で降順ソート
Ks = sorted(K, reverse = True)

w = 0
k = 0

for i in range(3) :
    w += Ws[i]
    k += Ks[i]

print(w, k)
```

(46) 未提出者は誰だ？ (Volume5_0511)

```
A = [i for i in range(1, 31)]
N = []

for i in range(28) :
    n = int(input())
    N.append(n)

R = list(set(A) - set(N))
# Nと重複していないAの要素をリスト化

print(min(R))
print(max(R))
```

(47) 集会所 (Volme4_0407)

```
n = int(input())
X = [int(x) for x in input().split()]

sub = max(X) - min(X)

if sub % 2 == 0 :
    print(sub // 2)
else :
    print(sub // 2 + 1)
```

(48) 最大の和 (Volume5_0516)

```
while True :
    n, k = map(int, input().split())

    if n == 0 :
        break
    else :
        A = []
        S = []
        s = 0
        cnt = 0

        for i in range(n) :
            A.append(int(input()))

        for i in range(k) :
            s += A[i]
        S.append(s)

        for j in range(k, n) :
            s = S[cnt] - A[j - k] + A[j]
            S.append(s)
            cnt += 1

        print(max(S))
```

(49) 山の高さ (Volume0_0001)

```
M = []

for i in range(10) :
    M.append(int(input()))

Ms = sorted(M, reverse = True)

for j in range(3) :
    print(Ms[j])
```

(50) 最頻値 (Volume0_0028)

```
N = []
A = [0 for i in range(100)]

while True :
    try :
        n = int(input())
        N.append(n)
        A[n - 1] += 1

    except :
        break

m = max(A)

M = [i for i, x in enumerate(A) if x == m]
l = len(M)

for i in range(l) :
    print(M[i] + 1)
```

- `try except`... 例外処理
  - `try`内の処理を行う
  - エラー等で指定の処理ができないとき`except`内の処理を行う
- `enumerate(X)` ... `X`の要素と同時にその要素のインデックスを取得する

(51) ヒストグラム (Volume2_0219)

```
while True :
    n = int(input())

    if n == 0 :
        break
    else :
        C = [0 for i in range(10)]

        for i in range(n) :
            C[int(input())] += 1

        for j in range(10) :
            if C[j] == 0 :
                print("-")
            else :
                for i in range(C[j]) :
                    print("*", end = "")
                print()
```

文字と文字列

(52) 大文字変換 (Volume0_0020)

```
print(input().upper())
```

- X.upper() ... 文字列Xを全て大文字に変換

(53) 大文字と小文字の入れ替え (ITP1_8_A)

```
print(input().swapcase())
```

- X.upper() ... 文字列Xの大文字と小文字を入れ替える

(54) 文字のカウント (ITP1_8_C)

```
a = "abcdefghijklmnopqrstuvwxyz"
A = [0 for i in range(26)]

while True :
    try :
        s = input()

        for i in range(len(s)) :
            if s[i] in a or s[i].lower() in a :
                A[a.index(s[i].lower())] += 1
    except :
        break

for j in range(26) :
    print(a[j], ":", A[j])
```

- X.lower() ... 文字列Xを全て小文字に変換

(55) 数字の和 (ITP1_8_B)

```
while True :
    n = input()

    if n == "0" :
        break
    else :
        s = 0
```

```
        for i in range(len(n)) :
            s += int(n[i])

        print(s)
```

(56) KUPC (Volume22_2271)

```
n = input()
KUPC = ["K", "U", "P", "C"]
kupc = [0 for i in range(4)]

for i in range(len(n)) :
    if n[i] in KUPC :
        kupc[KUPC.index(n[i])] += 1

print(min(kupc))
```

(57) 単語の検索 (ITP1_9_A)

```
W = input()
cnt = 0

while True :
    t = input()

    if t == "END_OF_TEXT" :
        break
    else :
        T = t.lower()
        T = T.split()

        for i in range(len(T)) :
            if T[i] == W :
                cnt += 1

print(cnt)
```

(58) リング (ITP1_8_D)

```
s = input()
p = input()

S = s * 2

if p in S :
    print("Yes")
```

```
else :
    print("No")
```

(59) 文字列の探索 (ALDS1_14_A)

```
T = input()
P = input()
l = len(P)

for i in range(len(T) - l + 1) :
    if T[i : i + l] == P :
        print(i)
```

(60) シャッフル (ITP1_9_B)

```
while True :
    C = input()

    if C == "-" :
        break
    else :
        m = int(input())

        for i in range(m) :
            h = int(input())
            c = C[0 : h]
            C = C[h : len(C) + 1]
            C += c

        print(C)
```

(61) カードゲーム (ITP1_9_C)

```
n = int(input())

t = 0
h = 0

for i in range(n) :
    T, H = map(str, input().split())

    if T > H :
        t += 3
    elif T < H :
        h += 3
    else :
```

```
        t += 1
        h += 1

print(t, h)
```

(62) バドミントン (Volume1_0174)

```
while True :
    d1 = input()

    if d1 == "0" :
        break
    else :
        d2 = input()
        d3 = input()

        d1A = d1[1 : len(d1)].count("A")
        d1B = len(d1) - d1A - 1
        if d2[0] == "A" :
            d1A += 1
        else :
            d1B += 1

        d2A = d2[1 : len(d2)].count("A")
        d2B = len(d2) - d2A - 1
        if d3[0] == "A" :
            d2A += 1
        else :
            d2B += 1

        d3A = d3[1 : len(d3)].count("A")
        d3B = len(d3) - d3A - 1
          if d3A > d3B :
              d3A += 1
          else :
              d3B += 1

        print(d1A, d1B)
        print(d2A, d2B)
        print(d3A, d3B)
```

いよいよITP入門からの卒業

(63)距離 (ITP1_10_A)

```
import math
```

```
x1, y1, x2, y2 = map(float, input().split())

x = (x2 - x1) ** 2
y = (y2 - y1) ** 2

print(f"{math.sqrt(x + y):.5f}")
```

- `math.sqrt(x)` ... $\sqrt{x}$

(64) 三角形 (ITP1_10_B)

```
import math

a, b, C = map(int, input().split())

rC = math.radians(C)
a2 = a ** 2
b2 = b ** 2
ab = a * b

c = math.sqrt(a2 + b2 - 2 * ab * math.cos(rC))

S = ab * math.sin(rC) / 2
L = a + b + c
h = b * math.sin(rC)

print(f"{S:.5f} {L:.5f} {h:.5f}")
```

(65) 標準偏差 (ITP1_10_C)

```
import math

while True :
    n = int(input())

    if n == 0 :
        break
    else :
        S = [int(s) for s in input().split()]
        a2 = 0

        m = sum(S) / n

        for i in range(n) :
            a2 += (S[i] - m) ** 2

        a = math.sqrt(a2 / n)

    print(f"{a:.5f}")
```

(66) 成績 (ITP1_7_A)

```
while True :
    m, f, r = map(int, input().split())

    if m == -1 and f == -1 and r == -1 :
        break
    else :
        if m == -1 or f == -1 or m + f < 30 :
            print("F")
        elif m + f >= 80 :
            print("A")
        elif m + f >= 65 and m + f < 80 :
            print("B")
        elif m + f >= 50 and m + f < 65 :
            print("C")
        else :
            if r >= 50 :
                print("C")
            else :
                print("D")
```

(67) 組み合わせの数 (ITP1_7_B)

```
while True :
    n, x = map(int, input().split())

    if n == 0 :
        break
    else :
        cnt = 0

        for i in range(1, n - 1) :
            for j in range(i + 1, n) :
                for k in range(j + 1, n + 1) :
                    if i + j + k == x :
                        cnt += 1

        print(cnt)
```

(68) 不足しているカードの発見 (ITP1_6_B)

```
n = int(input())

S = [i for i in range(1, 14)]
H = [i for i in range(1, 14)]
```

```
C = [i for i in range(1, 14)]
D = [i for i in range(1, 14)]

for i in range(n) :
    m, r = map(str, input().split())

    if m == "S" :
        S.remove(int(r))
    elif m == "H" :
        H.remove(int(r))
    elif m == "C" :
        C.remove(int(r))
    else :
        D.remove(int(r))

if len(S) > 0 :
    for i in range(len(S)) :
        print("S", S[i])
if len(H) > 0 :
    for i in range(len(H)) :
        print("H", H[i])
if len(C) > 0 :
    for i in range(len(C)) :
        print("C", C[i])
if len(D) > 0 :
    for i in range(len(D)) :
        print("D", D[i])
```

- `X.remove(y)` ... `y`と同値の要素を`X`から削除する
    - `y`と同値の要素が複数ある場合、一番初めの要素一つのみを削除する

(69) ベクトル積 (ITP1_6_D)

```
n, m = map(int, input().split())

A = []
B = []

for i in range(n) :
    A.append([int(a) for a in input().split()])

for j in range(m) :
    B.append(int(input()))

C = []

for i in range(n) :
    c = 0

    for j in range(m) :
        c += A[i][j] * B[j]
```

```
    C.append(c)

for i in range(n) :
    print(C[i])
```

(70) 行列積 (ITP1_7_D)

```
n, m, l = map(int, input().split())

A = []
B = []

for i in range(n) :
    A.append([int(a) for a in input().split()])

for j in range(m) :
    B.append([int(b) for b in input().split()])

C = []
L = []
s = 0

for i in range(l) :
    for j in range(n) :
        for k in range(m) :
            s += A[j][k] * B[k][i]

        L.append(s)
        s = 0

    C.append(L[i * n : i * n + n])

for i in range(n) :
    for j in range(l) :
        if j == l - 1 :
            print(C[j][i])
        else :
            print(C[j][i], end = " ")
```

(71) 表計算 (ITP1_7_C)

```
r, c = map(int, input().split())

A = []

for i in range(r) :
    A.append([int(a) for a in input().split()])
    A[i].append(sum(A[i]))
```

```
S = []

for i in range(c + 1) :
    s = 0

    for j in range(r) :
        s += A[j][i]

    S.append(s)

A.append(S)

for i in range(r + 1) :
    for j in range(c + 1) :
        if j == c :
            print(A[i][j])
        else :
            print(A[i][j], end = " ")
```

(72) ミンコフスキー距離 (ITP1_10_D)

```
import math

n = int(input())
X = [int(x) for x in input().split()]
Y = [int(y) for y in input().split()]

d1 = 0
d2 = 0
d3 = 0
Di = []

for i in range(n) :
    d1 += abs(X[i] - Y[i])
print(f"{d1:.6f}")

for i in range(n) :
    d2 += abs(X[i] - Y[i]) ** 2
print(f"{math.sqrt(d2):.6f}")

for i in range(n) :
    d3 += abs(X[i] - Y[i]) ** 3
print(f"{math.pow(d3, 1/3):.6f}")

for i in range(n) :
    Di.append(abs(X[i] - Y[i]))
print(f"{max(Di):.6f}")
```

- `pow(x, y)` ... x^y

整数と数値計算

(73) 素因数分解 (NTL_1_A)

```
n = int(input())
nn = n
N = []

print(n, ":", sep = "", end = " ")

if n % 2 == 0 :
    while n % 2 == 0 :
        N.append(2)
        n //= 2

for i in range(3, int(nn ** (1 / 2)) + 1, 2) :
    while n % i == 0 :
        N.append(i)
        n //= i

if n != 1 :
    N.append(n)

if len(N) == 0 :
    print(n)
else :
    for i in range(len(N)) :
        if i == len(N) - 1 :
            print(N[i])
        else :
            print(N[i], sep = "", end = " ")
```

- 整数Nの素因数の最大値は高々$\sqrt{x}$以下になるという性質を利用
    - $\sqrt{x}$を超えることが全くないというわけではないため、条件分岐で例外処理を行う

(74) べき乗 (NTL_1_B)

```
m, n = map(int, input().split())
mod = 1000000007

print(pow(m, n, mod))
```

- `pow(x, y, z)` ... $x^y mod z$
    - x^yをzで割った余り
    - `pow(x, y) % z`よりも処理が高速になる

(75) ユークリッドの互除法 (Volume1_0197)

```
while True :
    a, b = map(int, input().split())

    if a == 0 :
        break
    else :
        if a <= b :
                X, Y = b, a
        else :
            X, Y = a, b

        cnt = 0

        def Euc(X, Y, cnt) :
            while Y != 0 :
                cnt += 1
                X %= Y
                X, Y = Y, X

            return X, cnt

        print(Euc(X, Y, cnt)[0], Euc(X, Y, cnt)[1])
```

- `X, Y = a, b` ... 変数への代入は複数同時に行うこともできる
- `return x, y` ... `return`文では複数の値を返すことができる
    - 返り値が複数の場合はタプル型で返される
- タプル型 ... 読み取り専用の配列
- ユークリッドの互除法 ... *Euclidean Algorithm*

(76) 最小公倍数 (NTL_1_C)

```
import math

n = int(input())
A = [int(a) for a in input().split()]

ans = A[0]

for i in range(1, n) :
    ans = ans * A[i] // math.gcd(ans, A[i])

print(ans)
```

- LCM(Least Common Multiple) ... 最小公倍数
- GCD(Greatest Common Divisor) ... 最大公約数
- `math.gcd(x, y)` ... `x`と`y`の最大公約数を返す
    - `import math`が必要
- 二数の最小公倍数 ... $lcm(a, b) = \frac {a * b}{gcd(a, b)}$

- N個の数の最大公約数
    - 例　`9, 12, 6`の最大公約数は`3`
        - `gcd(9, 12) = 3`,`gcd(3, 6) = 3`と２つずつ順番に求める
- N個の最小公倍数
    - 二数の最小公倍数、N個の数の最大公約数の求め方を組み合わせる

(77) 最大公約数 (ALDS1_1_B)

```
import math

x, y = map(int, input().split())

print(math.gcd(x, y))
```

(78) 素数 (ALDS1_1_C)

```
import math

n = int(input())
cnt = 0

def prime(a) :
    for i in range(2, int(math.sqrt(a)) + 1) :
        if a % i == 0 :
            return False

    return True

for i in range(n) :
    if prime(int(input())) :
        cnt += 1

print(cnt)
```

- 素数 ... *Prime number*
- 素数判定 ... nが素数かどうかの確認は$\sqrt{n}$以下の数で割れば十分である

(79) 素数の数 (Volume0_0009)

```
import math

P = []
# 素数一覧リスト
Max = 1000000

def Era(P, Max) :
    D = list(range(2, Max))
```

```
    limit = math.sqrt(Max)
    while True :
        p = D[0]

        if limit <= p :
            P += D
            return P

        P.append(p)
        D = [d for d in D if d % p != 0]
        # D に含まれる値かつ p で割り切れない値をリスト化

Era(P, Max)

while True :
    try :
        n = int(input())

        if n == 1 :
            print(0)
        elif n == 2 :
            print(1)
        elif n >= max(P) :
            print(len(P))
        else :
            for i in range(n + 1) :
                if P[i] == n :
                    print(i + 1)
                    break
                elif P[i] > n :
                    print(i)
                    break

    except :
        break
```

- エラトステネスの篩（*Eratosthenes Sieve*）
    1. 2からnまでの整数を昇順に並べたリストDを作る
    2. D[0](リスト内で一番小さい値)をpとおく
    3. pを除くpの倍数を全てPから消していく
    4. pを素数リストPに追加する
    5. pが$\sqrt{n}$を超えたら終了
        - nが合成数であれば、必ず$\sqrt{n}$より小さい素因数を持つ
    6. P及び残ったDがn以下の全ての素数

(80) ゴールドバッハの予想 (Volume12_1200)

```
import math

P = []
```

```
# 素数一覧リスト
Max = 1000000

def Era(P, Max) :
    D = list(range(2, Max))
    limit = math.sqrt(Max)
    while True :
        p = D[0]

        if limit <= p :
            P += D
            return P

        P.append(p)
        D = [d for d in D if d % p != 0]

def Sub(x) :
    return abs(x - n)

def Sub2(x) :
    return abs(x - n // 2)

def Near(List, num):
# Listからnumに最も近い値を返す

    Ind = list(map(Sub, List))
    ind = min(Ind)
    # Listの要素とnumの差分を計算し最小値のインデックスを取得
    return Ind.index(ind)

def Near2(List, num):

    Ind = list(map(Sub2, List))
    ind = min(Ind)
    return Ind.index(ind)

Era(P, Max)

while True :
    n = int(input())

    if n == 0 :
        break
    else :
        l = Near2(P, n // 2) + 1
        N = Near(P, n) + 1
        cnt = 0
        s = 0
        I = []

        for i in range(N + 1, l - 2, -1) :
            for j in range(s, l + 1) :
                if n == P[i] + P[j] :
                    if P[j] not in I :
```

```
                    # 重複回避
                        cnt += 1
                        s = j
                        I.append(P[i])

                elif n < P[i] + P[j] :
                    break

        print(cnt)
```

！注意！ AOJでは`import numpy`は`Runtime Error`となる

`numpy`が使えた場合

```
# def Sub ~ def Near2 部分と入れ替え

import numpy as np

def Near(List, num):
# listからnumに最も近い値を返す

    index = abs(np.asarray(List) - num).argmin()
    # Listの要素とnumの差分を計算し最小値のインデックスを取得
    return index
```

- NumPyモジュール ... 多次元配列を扱う数値演算ライブラリ
- `np.asarray(X)` ... 引数に指定したリストを配列にする
- `X.argmin()` ... 配列`X`の最小値のインデックスを返す
    - 最小値が複数あったときは、インデックスの一番小さい値を返す

(81) コラッツの予想 (Volume1_0158)

```
while True :
    n = int(input())

    if n == 0 :
        break
    else :
        cnt = 0

        while True:
            if n == 1 :
                print(cnt)
                break

            elif n % 2 == 0 :
                n /= 2
                cnt += 1
```

```
            else :
                n *= 3
                n += 1
                cnt += 1
```

(82) 数値積分(Volume0_0014)

```
while True :
    try :
        d = int(input())
        l = 600 // d

        D = []

        for i in range(l - 1) :
            D.append(((len(D) + 1) ** 2) * d ** 3)

        print(sum(D))

    except :
        break
```

(83) 連立方程式 (Volume0_0004)

```
while True :
    try :
        a, b, c, d, e, f = map(float, input().split())

        x = (c * e - b * f) / (a * e - b * d)
        y = (c * d - a * f) / (b * d - a * e)

        if x == -0 :
            x = 0
        if y == -0 :
            y = 0

        print(f"{x:.3f} {y:.3f}")

    except :
        break
```

(84) 3乗根 (Volume0_0080)

```
while True :
    q = int(input())
```

```
    if q == -1 :
        break

    else :
        x = q / 2

        while abs(x ** 3 - q) >= 0.00001 * q :
            x -= ((x ** 3 - q) / (3 * x ** 2))

        print(f"{x:.6f}")
```

(85) 根の個数 (Volume22_2220)

```
import math

n = int(input())

def F(a, b, c, d) :
    return lambda x : a * x**3 + b * x**2 + c * x + d

def X(a, b, c) :
    a = 3 * a
    b = 2 * b

    try :
        D = math.sqrt(b**2 - 4 * a * c)

        if D == 0 :
            return 0
        else :
            return (-b + D) / (2 * a), (-b - D) / (2 * a)

    except :
        return -1

def boa(fz, lmax, lmin, p, n) :
    if fz > 0 :
        if lmin > 0 :
            p, n = 2, 1
        else :
            n = 3
    elif fz == 0 :
        if 0 < lmin and lmax < 0 :
            p, n = 1, 1
        elif lmax > 0 :
            p = 2
        else :
            n = 2
    else :
        if lmax > 0 :
            p = 3
```

```
            else :
                p, n = 1, 2

        return p, n

    def ao(fz, lmax, p, n) :
        if fz > 0 :
            n = 2
        elif fz == 0 :
            if lmax == 0 :
                p = 1
            else :
                n = 1
        else :
            if lmax > 0 :
                p = 2
            else :
                p, n = 1, 1

        return p, n

    def bo(fz, lmin, p, n) :
        if fz > 0 :
            if lmin > 0 :
                p, n = 1, 1
            else :
                n = 2
        elif fz == 0 :
            if lmin == 0 :
                n = 1
            else :
                p = 1
        else :
            p = 2

        return p, n

    def aob(fz, p, n) :
        if fz > 0 :
            n = 1
        elif fz < 0 :
            p = 1

        return p, n

    def P(x, y) :
        if x > 0 :
            print(y, 0)
        elif x < 0 :
            print(0, y)
        else :
            print(0, 0)

    for i in range(n) :
```

```
    a, b, c, d = map(int, input().split())
    if a < 0 :
        a *= -1
        b *= -1
        c *= -1
        d *= -1

    f = F(a, b, c, d)

    p = 0
    # Positive integer
    n = 0
    # Negative integer

    if X(a, b, c) == -1 :
        P(-d, 1)

    elif X(a, b, c) == 0 :
        if f(-b / (3 * a)) == 0 :
            P(-b, 3)
        else :
            P(-d, 1)

    else :
        if f(X(a, b, c)[0]) < f(X(a, b, c)[1]) :
            lmax = X(a, b, c)[1]
            # Local maximum
            lmin = X(a, b, c)[0]
            # Local minimum
        else :
            lmax = X(a, b, c)[0]
            lmin = X(a, b, c)[1]

        fmax = f(lmax)
        fmin = f(lmin)
        fz = f(0)

        if lmax == lmin :
            if lmax < 0 :
                print(0, 3)
            elif lmax > 0 :
                print(3, 0)
            else :
                print(0, 0)

        elif fmin < 0 and 0 < fmax :
            pn = boa(fz, lmax, lmin, p, n)
            print(pn[0], pn[1])

        elif fmax == 0 :
            pn = ao(fz, lmax, p, n)
            print(pn[0], pn[1])

        elif fmin == 0 :
```

```
            pn = bo(fz, lmin, p, n)
            print(pn[0], pn[1])

        else :
            pn = aob(fz, p, n)
            print(pn[0], pn[1])
```

文字列に慣れよう

(86) サーチエンジン (Volume0_0084)

```
import re

I = input().split()
S = []

for i in range(len(I)) :
    I[i] = re.split("[,.]", I[i])
    if 2 < len(I[i][0]) and len(I[i][0]) < 7 :
        S.append(I[i][0])

for i in range(len(S)) :
    if i == len(S) - 1 :
        print(S[i])
    else :
        print(S[i], " ", sep = "", end = "")
```

(87) 文字列反転 (Volume0_0006)

```
S = input()

cnt = -1

for i in range(len(S)) :
    if i == len(S) - 1 :
        print(S[0])
    else :
        print(S[cnt], sep = "", end = "")
        cnt -= 1
```

(88) Yes, I have a number (Volume10_1042)

```
while True :
    S = input()
```

```
    if S == "END OF INPUT" :
        break

    else :
        S += " "
        N = []
        s = 0
        b = 0

        for i in range(len(S)) :
            if S[i] != " " :
                s += 1
            else :
                b += 1
                if b == 1 :
                    N.append(s)
                    s = 0
                    b = 0
                else :
                    N.append(0)
                    b = 0

        for i in range(len(N)) :
            if i == len(N) - 1 :
                print(N[i])
            else :
                print(N[i], sep = "", end = "")
```

(89) CamelCase (Volume10_1044)

```
while True :
    name, type = map(str, input().split())

    if type == "X" :
        break

    else :
        N = list(name)

        if "_" in N :
            if type == "U" or type == "L" :
                cnt = 0

                for i in range(len(N)) :
                    if i == 0 and type == "U" :
                        N[i] = N[i].upper()
                    elif N[i] == "_" :
                        cnt += 1
                    elif cnt == 1 :
                        N[i] = N[i].upper()
                        cnt = 0
```

```
                N = [i for i in N if i != "_"]

        elif type == "U" :
            N[0] = N[0].upper()

        elif type == "L" :
            N[0] = N[0].lower()

        else :
            s = 0
            for i in range(len(N)) :
                if i == 0 :
                    N[s] = N[s].lower()
                    s += 1
                elif N[s].isupper() :
                    N[s] = N[s].lower()
                    N.insert(s, "_")
                    s += 2
                else :
                    s += 1

        for i in range(len(N)) :
            if i == len(N) - 1 :
                print(N[i])
            else :
                print(N[i], end = "")
```

- str型はイミュータブルな型であるから、一部要素を変更するためにミュータブルな型であるlist型に変換した
- X.insert(i, "x") … リストの指定した位置に新しい要素を挿入する
    - 第一引数 … 挿入位置
    - 第二引数 … 挿入する値
- X.isupper() … Xが全て大文字ならTrueを返し、そうでない時はFalseを返す

(90) JOIとIOI (Volume5_0522)

```
while True :
    try :
        n = input()
        J = 0
        I = 0

        for i in range(len(n) - 2) :
            if n[i : i + 3] == "JOI" :
                J += 1
            elif n[i : i + 3] == "IOI" :
                I += 1

        print(J)
        print(I)
```

```
    except :
        break
```

(91) 回文 (Volume0_0063)

```
cnt = 0

while True :
    try :
        n = list(input())
        N = list(reversed(n))

        if n == N :
            cnt += 1

    except :
        print(cnt)
        break
```

- `reversed(X)` … Xの要素を逆順にしたリストを返す
  - 元のリストは変更されない非破壊的処理
  - そのまま`print`すると`type`を表示するため、リストに変換して要素を読み込む

(92) カイザー暗号 (Volume5_0512)

```
A = [chr(i) for i in range(65, 65 + 26)]
# アルファベット大文字一覧表

D = [chr(i) for i in range(68, 68 + 23)]
D.append("A")
D.append("B")
D.append("C")
# シーザー暗号変換後アルファベット大文字一覧表

n = input()
ANS = []

for i in range(len(n)) :
    ANS.append(A[D.index(n[i])])

for i in range(len(ANS)) :
    if i == len(ANS) - 1 :
        print(ANS[i])
    else :
        print(ANS[i], end = "")
```

(93) 英語の文章 (Volume0_0029)

```
S = input().split()

l = 0
n = 0
L = "a"
N = "a"

for i in range(len(S)) :
    if S.count(S[i]) > n :
        n = S.count(S[i])
        N = S[i]

for i in range(len(S)) :
    if len(S[i]) > l :
        l = len(S[i])
        L = S[i]

print(N, L)
```

(94) りんごともも (Volume0_0050)

```
s = input()
S = []

a = "apple"
p = "peach"
A = []
P = []

for i in range(len(s)) :
    S.append(s[i])

for i in range(5) :
    A.append(a[i])
    P.append(p[i])

for i in range(len(S)) :
    if S[i : i + 5] == A :
        S[i : i + 5] = P
    elif S[i : i + 5] == P :
        S[i : i + 5] = A

for i in range(len(S)) :
    if i == len(S) - 1 :
        print(S[i])
    else :
        print(S[i], sep = "", end = "")
```

(95) 秘密の番号 (Volume0_0064)

```
import re

ans = 0

while True :
    try :
        s = input()

        num = re.findall("[0-9]+", s)

        for i in range(len(num)) :
            ans += int(num[i])

    except :
        print(ans)
        break
```

- `re.findall('[0-9]+', s)` ... 第一引数の条件を満たす全ての部分文字列をリストにして返す
  - `[0-9]` ... 0 ~ 9
  - `X+` ... Xが一回以上
- 正規表現一覧

(96) Hit and Blow(Volume0_0025)

```
while True :
    try :
        A = list(input().split())
        B = list(input().split())

        hit = 0
        blow = 0

        for i in range(4) :
            if A[i] == B[i] :
                hit += 1

            elif B[i] in A :
                blow += 1

        print(hit, blow)

    except :
        break
```

プログラミング・コンテスト A問題

(97) 等しい合計点 (Volume11_1153)

```
while True :
    n, m = map(int, input().split())

    if n == 0 :
        break

    else :
        T = []
        H = []

        for i in range(n) :
            T.append(int(input()))

        for i in range(m) :
            H.append(int(input()))

        Ts = sorted(T)
        Hs = sorted(H)

        sumT = sum(Ts)
        sumH = sum(Hs)

        d = (sumT - sumH) / 2
        D = abs(d)

        f = 0
        # flag

        if d < 0 :
            for i in range(len(Ts)) :
                if (Ts[i] + D) in Hs :
                    print(Ts[i], int(Ts[i] + D))
                    f = 1
                    break
        else :
            for i in range(len(Ts)) :
                if (Ts[i] - D) in Hs :
                    print(Ts[i], int(Ts[i] - D))
                    f = 1
                    break

        if f != 1 :
            print(-1)
```

(98) 連続する整数の和 (Volume21_2197)

```
N = [i for i in range(1, 1001)]
```

```
while True :
    n = int(input())

    if n == 0 :
        break

    else :
        cnt = 0

        for i in range(1, (n // 2) + 2) :
            s = N[i - 1]
            l = 1

            for j in range(i + 1, (n // 2) + 3) :
                if s < n :
                    s += N[j - 1]
                    l += 1

                elif s == n and l > 1:
                    cnt += 1
                    break
                else :
                    break

        print(cnt)
```

(99) 税率変更 (☆) (Volume11_1192)

```
while True :
    x, y, s = map(int, input().split())
    if x == 0 :
        break

    else :
        S = s // 2 + 1
        # a < b と考える
        c = 0
        # 新税率の最大価格

        for i in range(1, s - 1) :
            a = i * (100 + x) // 100
            # 旧税率

            if(a <= S) :
                for j in range(1, s - 1) :
                    b = j * (100 + x) // 100

                    if(s == a + b) :
                        A = i * (100 + y) // 100
                        # 新税率
                        B = j * (100 + y) // 100
```

```
                        if(A + B > c) :
                            c = A + B

            else :
                break

        print(c)
```

(100) 太郎君の買物 (Volume16_1616)

```
while True :
    n, m = map(int, input().split())

    if n == 0 :
        break

    else :
        A = list(map(int, input().split()))
        s = 0

        As = sorted(A)

        for i in range(len(As) - 1) :
            for j in range(i + 1, len(As)) :
                if As[i] + As[j] <= m and As[i] + As[j] > s :
                    s = As[i] + As[j]

                elif As[i] + As[j] > m :
                    break

        if s == 0 :
            print("NONE")
        else :
            print(s)
```

(101) 所得格差 (Volume16_1624)

```
while True :
    n = int(input())

    if n == 0 :
        break

    else :
        A = list(map(int, input().split()))
        cnt = 0
```

```
        ave = sum(A) / n
        # 平均 average

        for i in range(len(A)) :
            if A[i] <= ave :
                cnt += 1

        print(cnt)
```

(102) チェビシェフの定理 (Volume11_1172)

```
P = [1 for i in range(250000)]
# 素数リスト(0 ~)

P[0] = 0
P[1] = 0

for i in range(2, 250000) :
    for j in range(2, 250000 // i) :
        P[i * j] = 0
        # 倍数はフラグを0に

while True :
    n = int(input())

    if n == 0 :
        break

    else :
        print(P[n + 1 : 2 * n + 1].count(1))
```

(103) アルデンテ (Volume24_2406)

```
N, T, E = map(int, input().split())
X = list(map(int, input().split()))

time = [i for i in range(T - E, T + E + 1)]

clock = -2

for i in range(len(X)) :
    for j in range(len(time)) :
        if time[j] % X[i] == 0 :
            clock = X.index(X[i])

print(clock + 1)
```

(105) ディリクレの算術級数定理 (Volume11_1141)

```
P = [1 for i in range(1000000)]
# 素数フラグリスト(0 ~)

P[0] = 0
P[1] = 0

for i in range(2, 1000000) :
    for j in range(2, 1000000 // i) :
        P[i * j] = 0
        # 倍数はフラグを0に

while True :
    a, d, n = map(int, input().split())

    if a == 0 :
        break

    else :
        A = [i for i in range(a, 1000000, d)]
        cnt = 0

        for i in range(len(A)) :
            if P[A[i]] == 1 :
                cnt += 1

            if cnt == n :
                print(A[i])
                break
```

シミュレーション

(106) 病院の部屋番号 (Volume2_0208)

```
octD = {"0":0, "1":1, "2":2, "3":3, "4":5, "5":7, "6":8, "7":9}

while True :
    n = int(input())

    if n == 0 :
        break

    else :
        N = list(format(n, "o"))
        # nを8進数に変換

        for i in range(len(N)) :
            print(octD[N[i]], end = "")
        print()
```

- 10進数→8進数 ... `oct(x)` or `format(x, "o")`
    - `oct(x)` ... プレフィックス`0o`の後に`x`の8進数変換が続く**文字列**を返す
        - プレフィックス ... 接頭語
    - `format(x, "o")` ... `x`を8進数の**文字列**に変換する