

## A 「ABC 072 A - Sandglass2」

$X > t$  のとき、 $X - t$  秒。そうでないとき 0 秒

if 分を使い条件分岐しよう。

max を使うと多少楽かも。

解答例

```
int main() {
    int X, t;
    cin >> X >> t;
    cout << max(X - t, 0) << endl;
    return 0;
}
```

## B 「ABC 055 A - Restaurant」

$$x = 800N, y = 200(N/15)$$

よって、

$$x - y = 800N - 200(N/15)$$

解答例

```
int f(int n) {
    return 800 * n - 200 * (n / 15);
}

int main() {
    int n;
    cin >> n;
    cout << f(n) << endl;
    return 0;
}
```

関数使う必要はあまりないけど…

簡単な問題で関数の使い方を覚えておいてほしい。

7月にやる再起関数でまた説明すると思います。

## C 「ABC 009 B - 心配性な富豪、ファミリーレストランに行く」

まず、配列を降順ソートする。

ここで、前から二番目の要素は二番目に高い料理じゃないことに注意。

(多分、問題文にも書いてたので間違えないと思う)

前から順番に隣り合う 2 要素の値を比較していき、最初に値が違う場所を見つけた時の小さいほうの値が答え。

解答例

```
int main() {
    int n;
    cin >> n;
    vector<int> a(n);
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    sort(a.begin(), a.end(), greater<int>());
    for (int i = 0; i < n - 1; i++) {
        if (a[i] != a[i + 1]) {
            cout << a[i + 1] << endl;
            break;
        }
    }
    return 0;
}
```

break 忘れると 3 番目に高い料理以降も出てくる。

問題文が長いやつ選んでごめんね。

## D 「ABC 035 A. 高橋くんと回文」

文字列  $S$  とそれを逆さにした文字列  $T$  を考える。

$S, T$  の  $i$  文字目が両方とも  $*$  以外で文字が一致しない場合回文ではない。

解答例

```
int main() {
    string s, t, ans = "YES";
    cin >> s;
    t = s;
    reverse(t.begin(), t.end());
    for (int i = 0; i < s.size(); i++) {
        if (s[i] != '*' && t[i] != '*' && s[i] != t[i]) {
            ans = "NO";
        }
    }
    cout << ans << endl;
    return 0;
}
```

文字列の真ん中で探索を打ち切ることができる。

## E 「ABC 003 C - AtCoder プログラミング講座」

見る順番によってレートが変わるので、見る順番は大事。

レート  $A, B, C$  の動画をこの順に見たとき、

本目	レート
1	$\frac{A}{2}$
2	$\frac{\left(\frac{A}{2} + B\right)}{2} = \frac{A}{4} + \frac{B}{2}$
3	$\frac{\left(\frac{A}{4} + \frac{B}{2} + C\right)}{2} = \frac{A}{8} + \frac{B}{4} + \frac{C}{2}$

なんとなくパターンを読めたと思う。

$i$  番目に見る動画のレートを  $a_i$  とすると、 $K$  個の動画を見た時のレート  $X$  は、

$$X = \sum_{i=1}^K \frac{a_i}{2^{K-i+1}}$$

となり、 $i$  が大きいときにレートの高い動画を見たほうが良いとわかる。

レートはどれも 0 より大きいので、 $N$  個の動画の内、 $1 \sim K$  番目にレートが高い動画を選んで、レートの低いものから順に見て行けばよい。

昇順ソートをしたとき、 $N - K + 1$  番目から  $N$  番目までを見よう。

解答例

```
int main() {
    int n, k;
    cin >> n >> k;
    double ans = 0;
    vector<double> r(n);
    for (int i = 0; i < n; i++) {
        cin >> r[i];
    }
    sort(r.begin(), r.end());
    for (int i = 0; i < k; i++) {
        ans += r[n - k + i];
        ans /= 2;
    }
    cout << fixed << setprecision(15) << ans << endl;
    return 0;
}
```

答えが小数になるので double 型を使う。

答えの桁数に注意。

~~前回の問題とほとんど同じとか言わない。~~

## F 「ARC 099 C - Minimization」

この問題の答えはどんな並びだとしても

$$\frac{(N-2)}{(K-1)} + 1$$

## G 「ABC 085 D - Katana Thrower」

振るのは振った時一番ダメージの与えられる刀一本でいい。

↑の剣を振った時のダメージより、投げた時のダメージが高い刀のみ投げたほうがいい。

ある攻撃回数における最大ダメージは値が大きいものから足していけば求まる。

投げる刀の数と種類が同じならば刀を投げる順番を変えても与えるダメージは同じ。

これを踏まえると以下の手順で答えが求まる。

- [1]  $a_i$ をソートして最大値を求める。 $(a_0$ とする。)
- [2]  $b_i$ をソートして、 $a_0$ より大きい値がなくなるか敵の $h$ がなくなるまで、値の高い刀から順に投げていき $h$ を減らす。投げた回数を一回ずつ記録する。
- [3] 投げ終わった時 $h$ が残っている場合は、あと何回 $a_0$ を振れば $h$ が0を下回るかを求める。
- [4] [2]で投げた回数と[3]で振る回数を足したものが答え。

解答例

```
int main() {
    long long int n, h, ans = 0;
    cin >> n >> h;
    vector<int> a(n);
    vector<int> b(n);
    for (int i = 0; i < n; i++) {
        cin >> a[i] >> b[i];
    }
    sort(a.begin(), a.end(), greater<int>());
    sort(b.begin(), b.end(), greater<int>());
    for (int i = 0; i < n; i++) {
        if (a[0] < b[i] && h > 0) {
            ans++;
            h -= b[i];
        }
        else {
            break;
        }
    }
    if (h > 0) {
        ans += ((h - 1) / a[0] + 1);
    }
    cout << ans << endl;
    return 0;
}
```