A 問題

レベルアップ

概要

- 攻撃力 A と防御力 D が与えられます。
- どちらかを 1 加算した後の両方の積として考えられる最大値を求めてください。

解法

- $(A+1)D$ と $A(D+1)$ のうち大きい方を出力します。
- 入出力関数や条件分岐を組み合わせます。

- もしも最大値や最小値を求めるライブラリ等があればコードを書きやすくなります。

B 問題

運動管理

概要

- L 分以上 H 分以下の範囲で運動する必要があります。
- 各 A_i に対して、$A_i < L$ なら $L - A_i$ を、$L \leqq A_i \leqq H$ なら 0 を、$A_i > H$ なら -1 を出力してください。

解法

- 問題 A のように入出力関数や条件分岐を組み合わせて解くことができます。
- この問題ではさらにループや再帰関数を用いて計算を行う必要があります。

C 問題

数列ゲーム

概要

- 高橋君と青木君が数列で遊びます。
- 最初に高橋君が要素の 1 つに丸を付けて、その後青木君が別の要素 1 つに丸を付けます。
- 2 つの丸に囲まれた範囲(両端含む)を取り出し、左から交互に高橋君、青木君の得点に加算します。
- 青木君が自分の得点が最大となるように丸をつける(複数あるならそれらのうち最も左に丸をつける)とき、高橋君が得られる最大得点を求めてください。
- $2 \leqq N \leqq 50$

青木君の行動について

- 青木君の行動について、例とともに確認します。
- 例: 数列が[1, -3, 3, 9, 1, 6]
- 高橋君が左から 2 番目の要素 -3 を選んだ場合について考えます。
- 青木君は、左から 1 番目、3 番目、4 番目、5 番目、6 番目の要素を選ぶことができます。
- それぞれの選び方において、残る数列は [1, -3], [-3, 3], [-3, 3, 9], [-3, 3, 9, 1], [-3, 3, 9, 1, 6] となり、青木君の得られる得点は順に -3, 3, 3, 4, 4 となります。
- 最終的に青木君は左から 5 番目の要素に丸を付けます。

解法

- 高橋君が丸を付ける場所を固定したとき、青木君の行動がわかるので、青木君がどこに丸を付け、結果どちらが何点得られるかは一意に定まります。
- 高橋君の得点を求めるには、高橋君がここに丸を付けた場合に何点得られるかをすべての要素について求めておいて、それらの最小値を出力すれば解くことができます。

注意事項

- この問題には負の値が登場します。
- 負の値の取り扱い（使用する変数が符号付きなのか符号無しなのかなど）に注意してください。

D 問題

語呂合わせ

概要

- 正の整数 $v_1, \ldots, v_N$と文字列 $w_1, \ldots, w_N$ が与えられます。
- 文字列 $s_1, \ldots, s_K$ をうまく定めて、数字 i を文字列 s_i に変換することによりどの整数 v_j も対応する文字列 w_j にできるようにしてください。
- $1 \leqq N \leqq 50$
- $1 \leqq |s_i| \leqq 3$
- $1 \leqq v_i \leqq 10^9$
- $1 \leqq |w_i| \leqq 27$

部分点解法

- 先に文字列 $s_1, \ldots, s_K$ を決めて、それらが条件を満たすか判定します。
- データセット 1 において最大 $(3^3+3^2+3)^3$ 通りの組み合わせを試すことになりますが、この大きさならば余り多くの時間をかけずに計算することができます。
- 一方、データセット 2 になってくると最大 $(26^3+26^2+26)^9$ 通りを試すことになってしまい、これでは間に合いません。

考察

- 文字列の長さが固定された場合どうなるでしょうか?
- 実は、$s_1, \ldots, s_K$ の長さが定まると、文字列 $s_1, \ldots, s_K$ が文字列 $w_1, \ldots, w_N$ のうちどの範囲に合致するかが一意に定まります。
- 範囲が定まれば、$s_1, \ldots, s_K$ は一意に定まります (定まらなかった、すなわち矛盾したらその長さの固定の仕方では答えが無いことが言えます)。

満点解法

- 文字列 $s_1, \ldots, s_K$ の長さを固定することで、O($\sum |w_i| \times K$)で答えを求めることができます。
- $s_1, \ldots, s_K$ の文字列長は 3 以下なので、全体の計算量は O($3^K \times \sum |w_i| \times K$) に抑えることができ、これならばデータセット 2 でも高速に動作します。