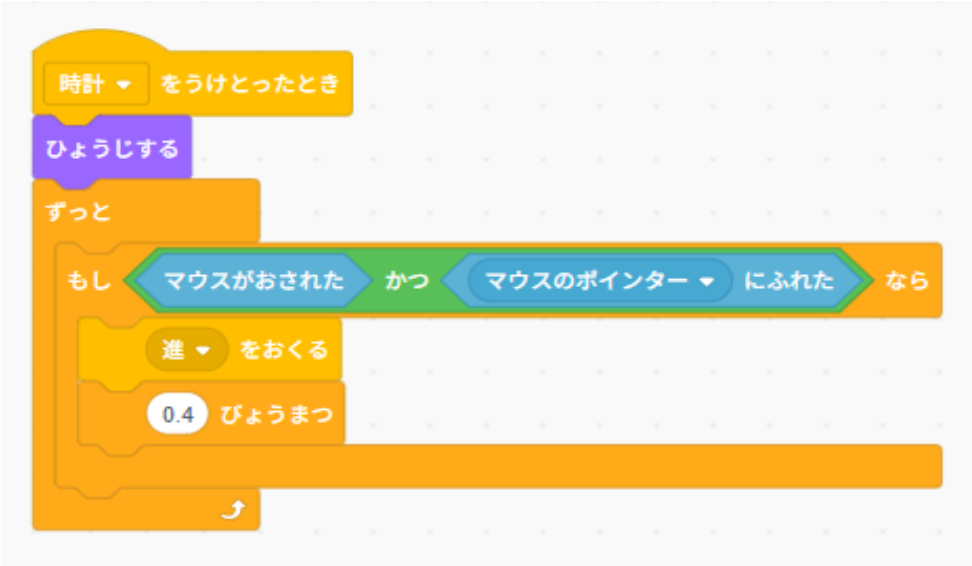
採点結果詳細【技術力】	
(1). 場合分け（例：もし）、もしくは 繰り返し（例：ずっと）、を使っている	10／10点

よくできているところ

正しく使えています。



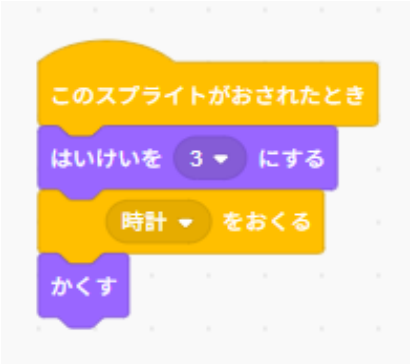
採点結果詳細【技術力】	
(1). 場合分け（例：もし）、もしくは 繰り返し（例：ずっと）、を使っている	
改善できそうなところ	特になし

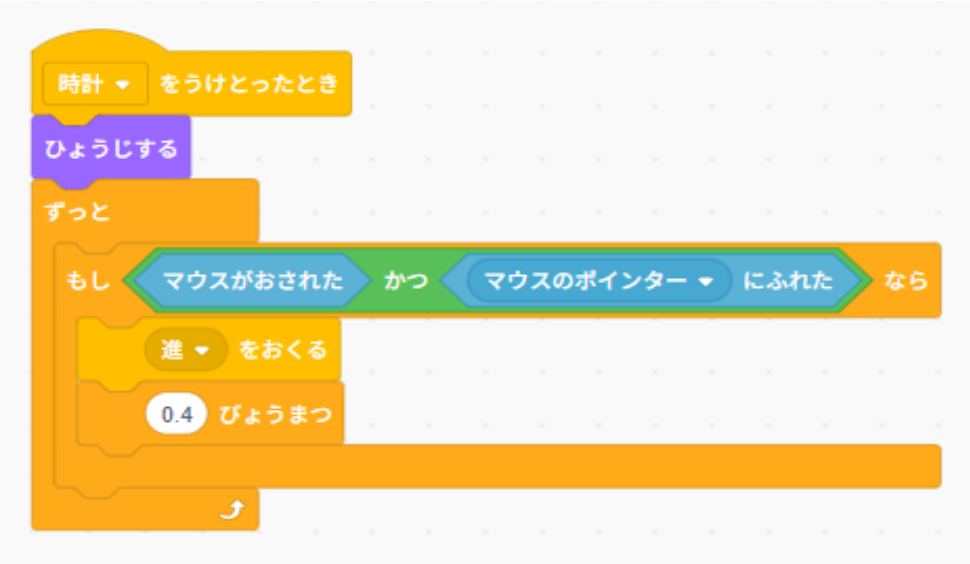
採点結果詳細【技術力】	
(1). 場合分け（例：もし）、もしくは 繰り返し（例：ずっと）、を使っている	
改善案	特になし

採点結果詳細【技術力】	
(2). メッセージを使っている	10／10点

よくできているところ

正しく使えています。





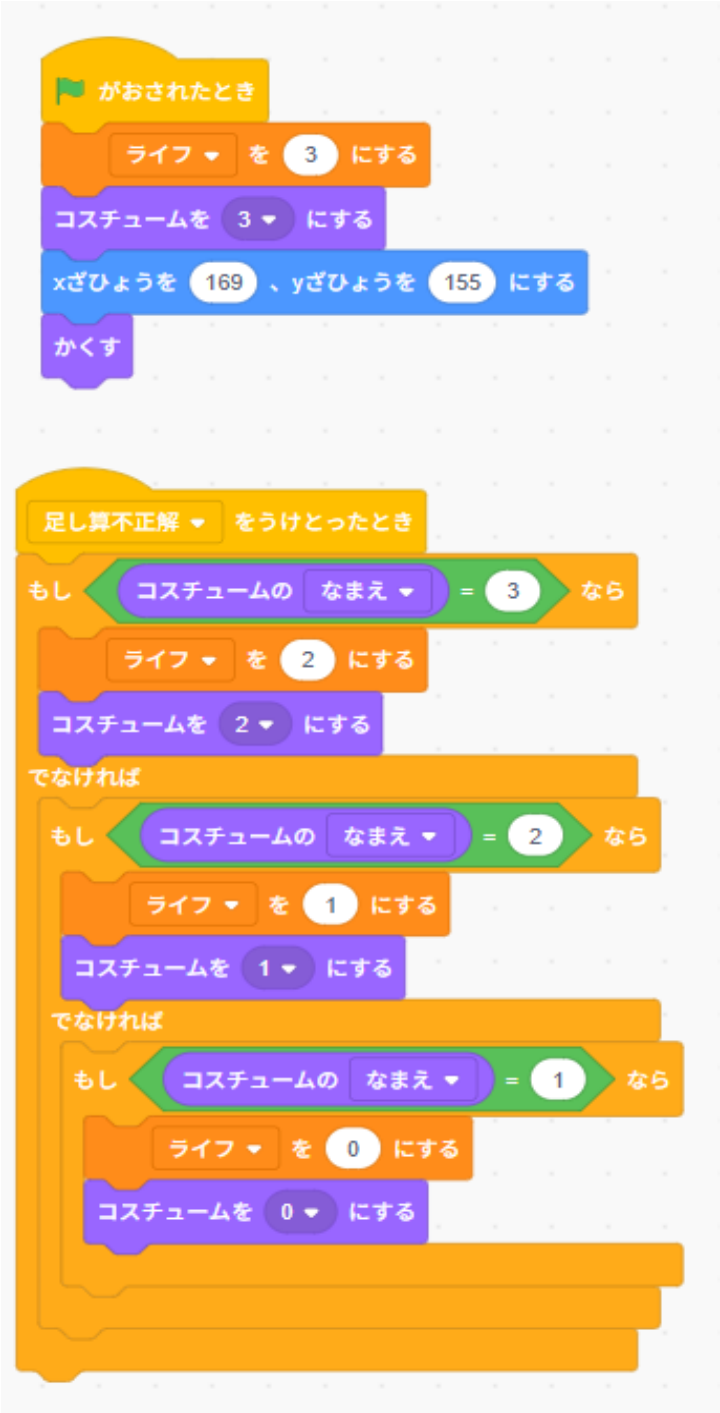
採点結果詳細【技術力】	
(2). メッセージを使っている	
改善できそうなところ	特になし

採点結果詳細【技術力】	
(2). メッセージを使っている	
改善案	特になし

採点結果詳細【技術力】	
(3). 変数を使っている	10／10点

よくできているところ

正しく使えています。



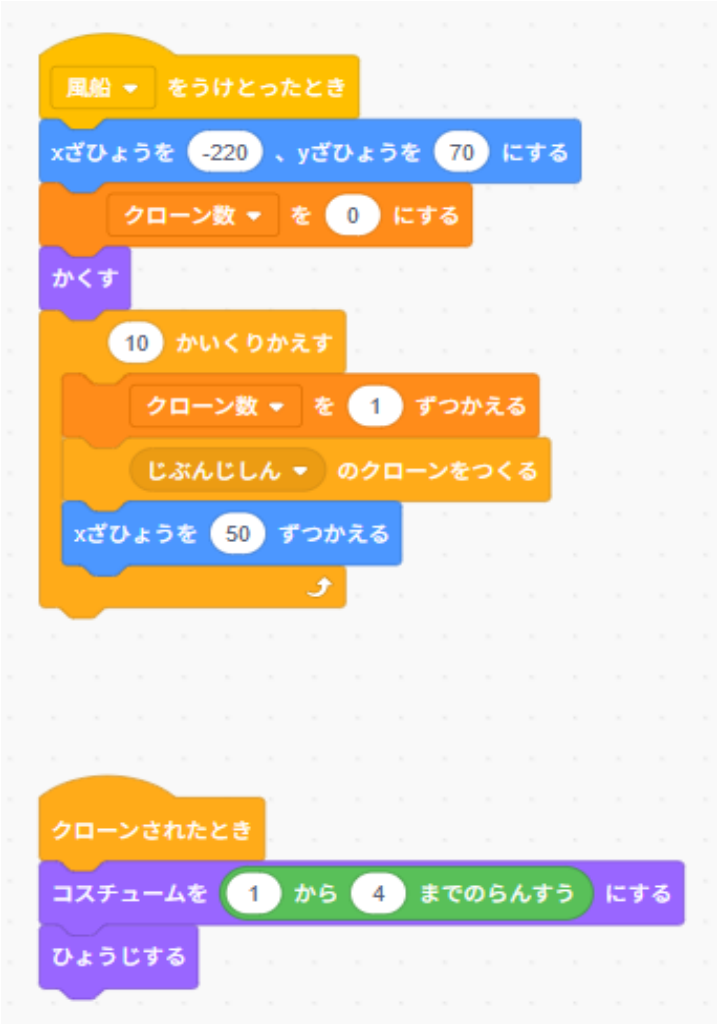
採点結果詳細【技術力】	
(3). 変数を使っている	
改善できそうなところ	特になし

採点結果詳細【技術力】	
(3). 変数を使っている	
改善案	特になし

採点結果詳細【技術力】	
(4). クローンを使っている	10／10点

よくできているところ

正しく使えています。



採点結果詳細【技術力】	
(4). クローンを使っている	
改善できそうなところ	特になし

採点結果詳細【技術力】	
(4). クローンを使っている	
改善案	特になし

採点結果詳細【技術力】	
(5). ブロック定義を使っている	5 ／10点

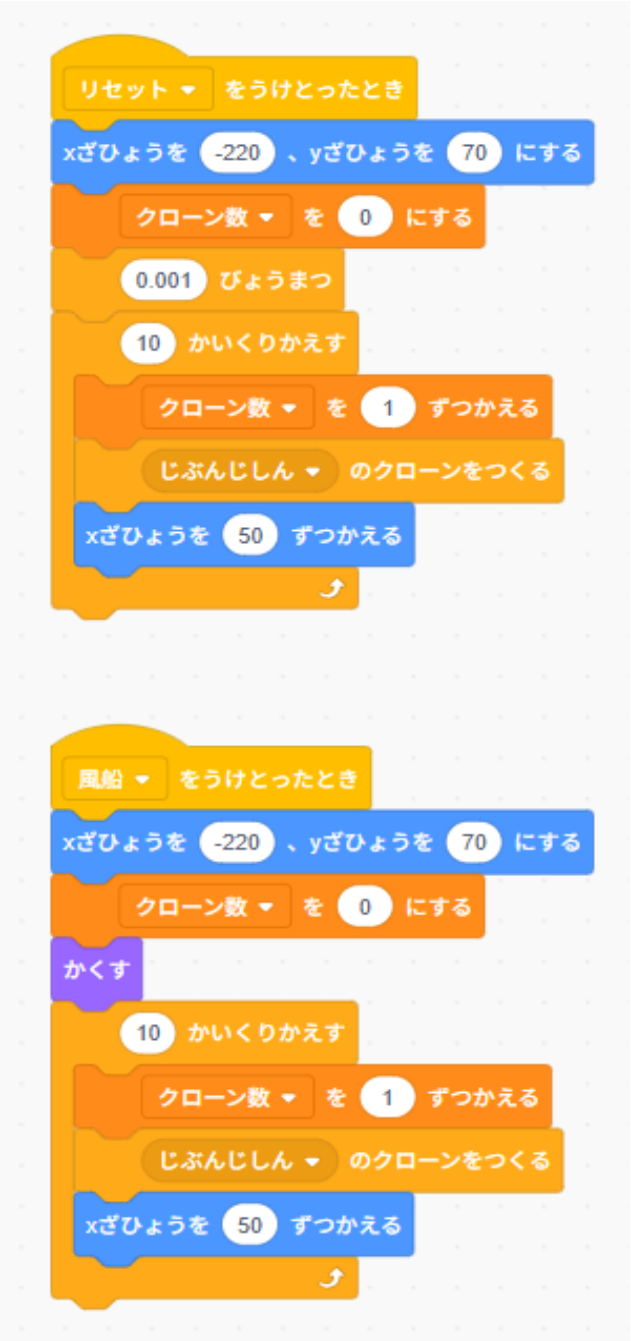
よくできているところ

正しく使えています。

採点結果詳細【技術力】
(5). ブロック定義を使っている

改善できそうなところ

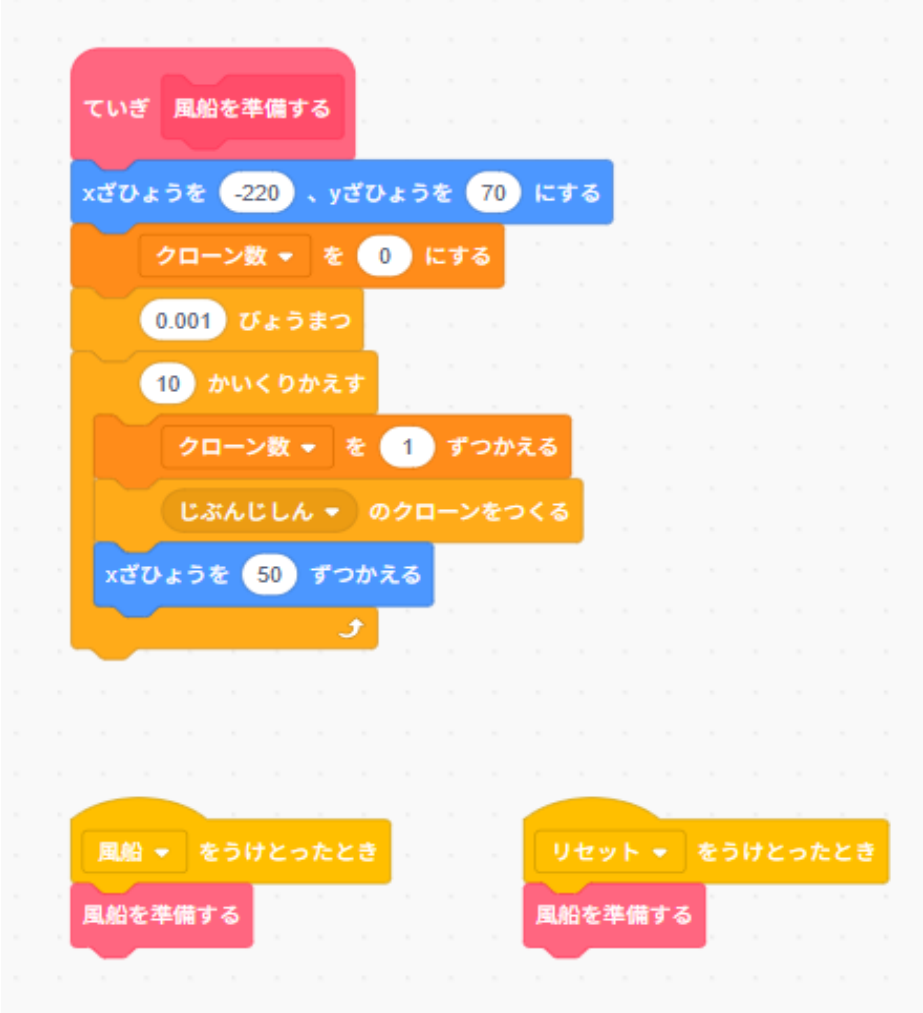
以下のプログラムはとてもよく似ています。



採点結果詳細【技術力】
(5). ブロック定義を使っている

改善案

ブロック定義を使って分かりやすいぷろぐらむにしましょう。



採点結果詳細【技術力】	
(6). リストを使っている	10／10点

よくできているところ

正しく使えています。

採点結果詳細【技術力】	
(6). リストを使っている	
改善できそうなところ	特になし

採点結果詳細【技術力】	
(6). リストを使っている	
改善案	特になし

採点結果詳細【技術力】	
(7). プログラムの品質 （１つの指摘につき３点減点）	4 ／10点

よくできているところ	
	プログラムの品質は全体的に高いですが、 まだ改善の余地はありそうです。

採点結果詳細【技術力】
(7). プログラムの品質 （１つの指摘につき３点減点）

改善できそうなところ

使われていないブロックがあります。

下記のプログラムは動作の順序が保証されていません。
右側のプログラムが先に実行される可能性があります。

採点結果詳細【技術力】
(7). プログラムの品質 （１つの指摘につき３点減点）

改善案

使わないブロックは削除しましょう。

メッセージ（送って待つ）を使って順序を正確に制御できます。