Erlang and message passing

What does process `Q` print?

process `P`

```
p() ->  % Q is Q's pid
  Q ! {self(), 0},
  Q ! {self(), 2}.
```

process `Q`

```
q() ->  % P is P's pid
  receive {P, N} ->
    io:format("~p", [N+1]) end,
  q().
```

1. 0 and 2, in any order
2. 0 and then 2
3. 1 and then 3
4. 1 and 3, in any order

What does process `Q` print?

process `P`

```
p() ->  % Q is Q's pid
  Q ! {self(), 0},
  Q ! {self(), 2}.
```

process `Q`

```
q() ->  % P is P's pid
  receive {P, N} ->
    io:format("~p", [N+1]) end,
  q().
```

1. 0 and 2, in any order
2. 0 and then 2
3. 1 and then 3
4. 1 and 3, in any order

What do processes `P` and `Q` print?

process `P`

```
p() ->  % Q is Q's pid
  Q ! {Q, 0},
  receive {P, N} ->
    io:format("~p", [N+1])
  end.
```

process `Q`

```
q() ->  % P is P's pid
  P ! {P,2},
  receive {Q, N} ->
    io:format("~p", [N+1])
  end.
```

1. 0 and 2, in any order
2. 0 and then 2
3. 1 and then 3
4. 1 and 3, in any order

What do processes `P` and `Q` print?

process `P`

```
p() ->  % Q is Q's pid
  Q ! {Q, 0},
  receive {P, N} ->
    io:format("~p", [N+1])
  end.
```

process `Q`

```
q() ->  % P is P's pid
  P ! {P,2},
  receive {Q, N} ->
    io:format("~p", [N+1])
  end.
```

1. 0 and 2, in any order
2. 0 and then 2
3. 1 and then 3
4. 1 and 3, in any order

What does process `Q` print?

process `P`

```
p() ->  % Q is Q's pid
  self() ! self(),
  receive self() ->
    Q !
    {self(),
     fun (Y) -> Y+1 end}
  end.
```

process `Q`

```
q() ->  % P is P's pid
  receive {P, F} ->
    io:format("~p", [F(3)]) end.
```

1. 3
2. 4
3. P's pid (process identifier)
4. Q's pid (process identifier)

What does process `Q` print?

process `P`

```
p() ->  % Q is Q's pid
  self() ! self(),
  receive self() ->
    Q !
    {self(),
     fun (Y) -> Y+1 end}
  end.
```

process `Q`

```
q() ->  % P is P's pid
  receive {P, F} ->
    io:format("~p", [F(3)]) end.
```

1. 3
2. 4
3. P's pid (process identifier)
4. Q's pid (process identifier)