Found Plan (output)

- (unstack c b)
- (stack c f)
- (unstack e d)
- (putdown e)
- (pickup d)
- (stack d c)
- (unstack b a)
- (putdown b)
- (pickup e)
- (stack e a)

```
(:action unstack
  :parameters (c b)
  :precondition
    (and
      (on c b)
      (clear c)
      (arm-empty)
    )
  :effect
    (and
      (holding c)
      (clear b)
      (not
        (on c b)
      )
      (not
        (clear c)
      )
      (not
        (arm-empty)
      )
    )
)
```

Found Plan (output)

- (unstack c b)
- (stack c f)
- (unstack e d)
- (putdown e)
- (pickup d)
- (stack d c)
- (unstack b a)
- (putdown b)
- (pickup e)
- (stack e a)

```
(:action stack
  :parameters (c f)
  :precondition
    (and
      (clear f)
      (holding c)
    )
  :effect
    (and
      (arm-empty)
      (clear c)
      (on c f)
      (not
        (clear f)
      )
      (not
        (holding c)
      )
    )
)
```

Found Plan (output)

- (unstack c b)
- (stack c f)
- (unstack e d)
- (putdown e)
- (pickup d)
- (stack d c)
- (unstack b a)
- (putdown b)
- (pickup e)
- (stack e a)

```
(:action unstack
  :parameters (e d)
  :precondition
    (and
      (on e d)
      (clear e)
      (arm-empty)
    )
  :effect
    (and
      (holding e)
      (clear d)
      (not
        (on e d)
      )
      (not
        (clear e)
      )
      (not
        (arm-empty)
      )
    )
)
```

Found Plan (output)

- (unstack c b)
- (stack c f)
- (unstack e d)
- (putdown e)
- (pickup d)
- (stack d c)
- (unstack b a)
- (putdown b)
- (pickup e)
- (stack e a)

```
(:action putdown
  :parameters (e)
  :precondition
    (and
      (holding e)
    )
  :effect
    (and
      (clear e)
      (arm-empty)
      (on-table e)
      (not
        (holding e)
      )
    )
)
```

Found Plan (output)

(unstack c b)

(stack c f)

(unstack e d)

(putdown e)

(pickup d)

(stack d c)

(unstack b a)

(putdown b)

(pickup e)

(stack e a)

```
(:action pickup
  :parameters (d)
  :precondition
    (and
      (clear d)
      (on-table d)
      (arm-empty)
    )
  :effect
    (and
      (holding d)
      (not
        (clear d)
      )
      (not
        (on-table d)
      )
      (not
        (arm-empty)
      )
    )
)
```

Found Plan (output)

(unstack c b)

(stack c f)

(unstack e d)

(putdown e)

(pickup d)

(stack d c)

(unstack b a)

(putdown b)

(pickup e)

(stack e a)

```
(:action stack
  :parameters (d c)
  :precondition
    (and
      (clear c)
      (holding d)
    )
  :effect
    (and
      (arm-empty)
      (clear d)
      (on d c)
      (not
        (clear c)
      )
      (not
        (holding d)
      )
    )
)
```

Found Plan (output)

(unstack c b)

(stack c f)

(unstack e d)

(putdown e)

(pickup d)

(stack d c)

(unstack b a)

(putdown b)

(pickup e)

(stack e a)

```
(:action unstack
  :parameters (b a)
  :precondition
    (and
      (on b a)
      (clear b)
      (arm-empty)
    )
  :effect
    (and
      (holding b)
      (clear a)
      (not
        (on b a)
      )
      (not
        (clear b)
      )
      (not
        (arm-empty)
      )
    )
)
```

Found Plan (output)

(unstack c b)

(stack c f)

(unstack e d)

(putdown e)

(pickup d)

(stack d c)

(unstack b a)

(putdown b)

(pickup e)

(stack e a)

```
(:action putdown
  :parameters (b)
  :precondition
    (and
      (holding b)
    )
  :effect
    (and
      (clear b)
      (arm-empty)
      (on-table b)
      (not
        (holding b)
      )
    )
)
```

Found Plan (output)

(unstack c b)

(stack c f)

(unstack e d)

(putdown e)

(pickup d)

(stack d c)

(unstack b a)

(putdown b)

(pickup e)

(stack e a)

```
(:action pickup
  :parameters (e)
  :precondition
    (and
      (clear e)
      (on-table e)
      (arm-empty)
    )
  :effect
    (and
      (holding e)
      (not
        (clear e)
      )
      (not
        (on-table e)
      )
      (not
        (arm-empty)
      )
    )
)
```

Found Plan (output)

(unstack c b)

(stack c f)

(unstack e d)

(putdown e)

(pickup d)

(stack d c)

(unstack b a)

(putdown b)

(pickup e)

(stack e a)

```
(:action stack
  :parameters (e a)
  :precondition
    (and
      (clear a)
      (holding e)
    )
  :effect
    (and
      (arm-empty)
      (clear e)
      (on e a)
      (not
        (clear a)
      )
      (not
        (holding e)
      )
    )
)
```