

Name: Maloth Aditya

Roll No.: 120CS0124

Q1.

Domain Code:

```
(define (domain trucking)

  (:requirements :strips :negative-preconditions :equality)

  (:predicates
    (CITY ?x)
    (TRUCK ?x)
    (PACKAGE ?x)
    (is-empty ?x)
    (at-city ?x ?y)
    (carry ?x ?y)
  )

  (:action load
    :parameters (?x ?y ?z)
    :precondition (and
      (PACKAGE ?x) (TRUCK ?y) (CITY ?z)
      (at-city ?x ?z)
      (at-city ?y ?z)
      (is-empty ?y)
    )
    :effect (and
      (not (is-empty ?y))
      (carry ?y ?x)
      ; (not (at-city ?x ?z))
    )
  )

  (:action unload
    :parameters (?x ?y ?z)
    :precondition (and
      (PACKAGE ?x) (TRUCK ?y) (CITY ?z)
      ; (not (at-city ?x ?z))
      (at-city ?y ?z)
      (not (is-empty ?y))
      (carry ?y ?x)
    )
  )
)
```

```
    :effect (and
      (is-empty ?y)
      (at-city ?x ?z)
    )
  )
  (:action drive
    :parameters (?x ?y ?z ?p1)
    :precondition (and
      (CITY ?x) (CITY ?y) (TRUCK ?z) (PACKAGE ?p1)
      (at-city ?z ?x)
      (carry ?z ?p1)
      (at-city ?p1 ?x)
      (not (at-city ?p1 ?y))
    )
    :effect (and
      (at-city ?z ?y)
      (not (at-city ?z ?x))
      (at-city ?p1 ?y)
      (not (at-city ?p1 ?x))
    )
  )
)
```

Problem Code:

```
(define (problem problem_name) (:domain trucking)
  (:objects truck1 truck2 package1 package2 package3 package4 package5 package6
    package7 package8 city1 city2)

  (:init
    (TRUCK truck1) (TRUCK truck2)
    (CITY city1) (CITY city2)
    (PACKAGE package1) (PACKAGE package2)
    ; (PACKAGE package3) (PACKAGE package4)
    (PACKAGE package5) (PACKAGE package6)
    ; (PACKAGE package7) (PACKAGE package8)
    (at-city package1 city1) (at-city package2 city1)
    ; (at-city package3 city1) (at-city package4 city1)
    (at-city package5 city2) (at-city package6 city2)
    ; (at-city package7 city2) (at-city package8 city2)
    (is-empty truck1) (is-empty truck2)
    (at-city truck1 city1)
    (at-city truck2 city2)
  )
)
```

AI Lab | 31 October 2023

```
(:goal (and
  (at-city package1 city2) (at-city package2 city2)
  ; (at-city package3 city2) (at-city package4 city2)
  (at-city package5 city1) (at-city package6 city1)
  ; (at-city package7 city1) (at-city package8 city1)
))
)
```

OUTPUT:

Planner output ✕

```
load package6 truck2 city2
drive city2 city1 truck2 package6
load package2 truck1 city1
drive city1 city2 truck1 package2
unload package2 truck1 city2
load package5 truck1 city2
drive city2 city1 truck1 package5
unload package5 truck1 city1
load package1 truck1 city1
drive city1 city2 truck1 package1
```

Q2.

Domain Code:

```
(define (domain blocksOnTable)
  (:requirements :strips :negative-preconditions :equality)

  (:predicates
    (BLOCK ?x)
    (TABLE ?x)
    (TOPFREE ?x)
    (ONTOPOF ?x ?y))
```

```
)

(:action moveBlockFromToTopOfTable
  :parameters (?x ?y ?z)
  :precondition (and
    (BLOCK ?x)
    (TABLE ?z)
    (TOPFREE ?x)
    (TOPFREE ?z)
    (ONTOPOF ?x ?y)
  )
  :effect (and
    (TOPFREE ?y)
    (not (ONTOPOF ?x ?y))
    (ONTOPOF ?x ?z)
  )
)

)

(:action moveBlockFromToTopOfBlock
  :parameters (?x ?y ?z)
  :precondition (and
    (BLOCK ?x)
    (BLOCK ?z)
    (TOPFREE ?x)
    (TOPFREE ?z)
    (ONTOPOF ?x ?y)
  )
  :effect (and
    (TOPFREE ?y)
    (not (TOPFREE ?z))
    (not (ONTOPOF ?x ?y))
    (ONTOPOF ?x ?z)
  )
)

)

)
```

Problem Code:

```
(define (problem arrangeBlocks) (:domain blocksOnTable)
(:objects blockA blockB blockC blockD table)

(:init
  (BLOCK blockA)
  (BLOCK blockB)
  (BLOCK blockC)
```

```
(BLOCK blockD)
(TABLE table)
(ONTOPOF blockA table)
(ONTOPOF blockC table)
(ONTOPOF blockB blockC)
(ONTOPOF blockD blockB)
(TOPFREE blockD)
(TOPFREE blockA)
(TOPFREE table)
)

(:goal (and
  (ONTOPOF blockD table)
  (ONTOPOF blockC blockD)
  (ONTOPOF blockB blockC)
  (ONTOPOF blockA blockB)
  (TOPFREE blockA)
))
)
```

OUTPUT:

Planner output ✕

moveBlockFromToTopOfTable blockD blockB table

moveBlockFromToTopOfBlock blockB blockC blockA

moveBlockFromToTopOfBlock blockC table blockD

moveBlockFromToTopOfBlock blockB blockA blockC

moveBlockFromToTopOfBlock blockA table blockB

Q3.

Domain Code:

```
(define (domain problem3)

(:requirements :negative-preconditions :equality)

(:predicates
  (LOCATION ?x)
  (MAGICALOBJECTS ?x)
  (HORCRUXES ?x)
  (ENEMY ?x)
  (HARRY ?x)
  (HAS ?x)
  (BROOM ?x)
  (WAND ?x)
  (FANG ?x)
  (SWORD ?x)
  (LOCKET ?x)
  (CUP ?x)
  (ROOMOFREQUIREMENTS ?x)
  (FORESTOFDEAN ?x)
  (CHAMBEROFSECRETS ?x)
  (OLIVANDERS ?x)
  (MINISTRYOFMAGIC ?x)
  (VAULT ?x)
  (FORBIDDENFOREST ?x)
  (LOCKETDESTROYED)
  (CUPDESTROYED)
  (ENEMYDEAD)
  (AT ?x)
  (DEAD ?x)
)

(:action KILLENNEMY
  :parameters (?x ?y ?z)
  :precondition (and
    (AT ?x)
    (FORBIDDENFOREST ?x)
    (not (DEAD ?z))
    (ENEMY ?z)
    (HAS ?y)
    (WAND ?y)
    (LOCKETDESTROYED)
    (CUPDESTROYED)
  )
  :effect (DEAD ?z)
)
```

```
(:action DESTROYLOCKET
  :parameters (?x ?y)
  :precondition (and
    (AT ?x)
    (MINISTRYOFMAGIC ?x)
    (not (LOCKETDESTROYED))
    (HAS ?y)
    (SWORD ?y)
  )
  :effect (LOCKETDESTROYED)
)

(:action DESTROYCUP
  :parameters (?x ?y)
  :precondition (and
    (AT ?x)
    (VAULT ?x)
    (not (CUPDESTROYED))
    (HAS ?y)
    (FANG ?y)
  )
  :effect (CUPDESTROYED)
)

(:action PICKUPWAND
  :parameters (?x ?y)
  :precondition (and
    (AT ?x)
    (OLIVANDERS ?x)
    (not (HAS ?y))
    (WAND ?y)
  )
  :effect (HAS ?y)
)

(:action PICKUPFANG
  :parameters (?x ?y)
  :precondition (and
    (AT ?x)
    (CHAMBEROFSECRETS ?x)
    (not (HAS ?y))
    (FANG ?y)
  )
  :effect (HAS ?y)
)
```

```
)

(:action PICKUPSWORD
  :parameters (?x ?y)
  :precondition (and
    (AT ?x)
    (FORESTOFDEAN ?x)
    (not (HAS ?y))
    (SWORD ?y)
  )
  :effect (HAS ?y)
)

(:action PICKUPBROOM
  :parameters (?x ?y)
  :precondition (and
    (AT ?x)
    (ROOMOFREQUIREMENTS ?x)
    (not (HAS ?y))
    (BROOM ?y)
  )
  :effect (HAS ?y)
)

(:action GOFROMTO
  :parameters (?x ?y ?z)
  :precondition (and
    (AT ?x)
    (not (AT ?y))
    (HAS ?z)
    (BROOM ?z)
  )
  :effect (and
    (AT ?y)
    (not (AT ?x))
  )
)

)
```

Problem Code:

```
(define (problem problem_name) (:domain problem3)
  (:objects sword fang wand harry broom locket cup forestofdean chamberofsecrets
    roomofrequirements olivanders enemy ministryofmagic vault forbiddenforest
```


AI Lab | 31 October 2023

```
)  
  
(:init  
  (BROOM broom)  
  (SWORD sword)  
  (WAND wand)  
  (FANG fang)  
  (ROOMOFREQUIREMENTS roomofrequirements)  
  (FORESTOFDEAN forestofdean)  
  (CHAMBEROFSECRETS chamberofsecrets)  
  (OLIVANDERS olivanders)  
  (MINISTRYOFMAGIC ministryofmagic)  
  (VAULT vault)  
  (FORBIDDENFOREST forbiddenforest)  
  (ENEMY enemy)  
  (AT roomofrequirements)  
)  
  
(:goal (DEAD enemy))  
)
```

OUTPUT:

Planner output X

- PICKUPBROOM roomofrequirements broom
- GOFROMTO roomofrequirements forestofdean broom
- GOFROMTO forestofdean chamberofsecrets broom
- PICKUPFANG chamberofsecrets fang
- GOFROMTO chamberofsecrets olivanders broom
- GOFROMTO olivanders ministryofmagic broom
- GOFROMTO ministryofmagic vault broom
- DESTROYCUP vault fang
- GOFROMTO vault forbiddenforest broom
- GOFROMTO forbiddenforest olivanders broom
- PICKUPWAND olivanders wand
- GOFROMTO olivanders forestofdean broom
- PICKUPSWORD forestofdean sword
- GOFROMTO forestofdean ministryofmagic broom
- DESTROYLOCKET ministryofmagic sword
- GOFROMTO ministryofmagic forbiddenforest broom
- KILLENEMY forbiddenforest wand enemy