

# BÁO CÁO ĐỒ ÁN

## LEGO MINDSTORMS

***Sinh viên thực hiện:***

Nguyễn Duy Khải – 1712513

Nguyễn Trọng Khiêm – 1712529

Nguyễn Hữu Huân – 1712466

***Giảng viên hướng dẫn:***

Nguyễn Ngọc Thảo

# BÁO CÁO ĐỒ ÁN 2

## 1. Thông tin nhóm

STT	MSSV	Họ và tên
1	1712529	Nguyễn Trọng Khiêm
2	1712513	Nguyễn Duy Khải
3	1712466	Nguyễn Hữu Huân

## 2. Chi tiết thực hiện

### 2.1. Hợp giải trên logic mệnh đề

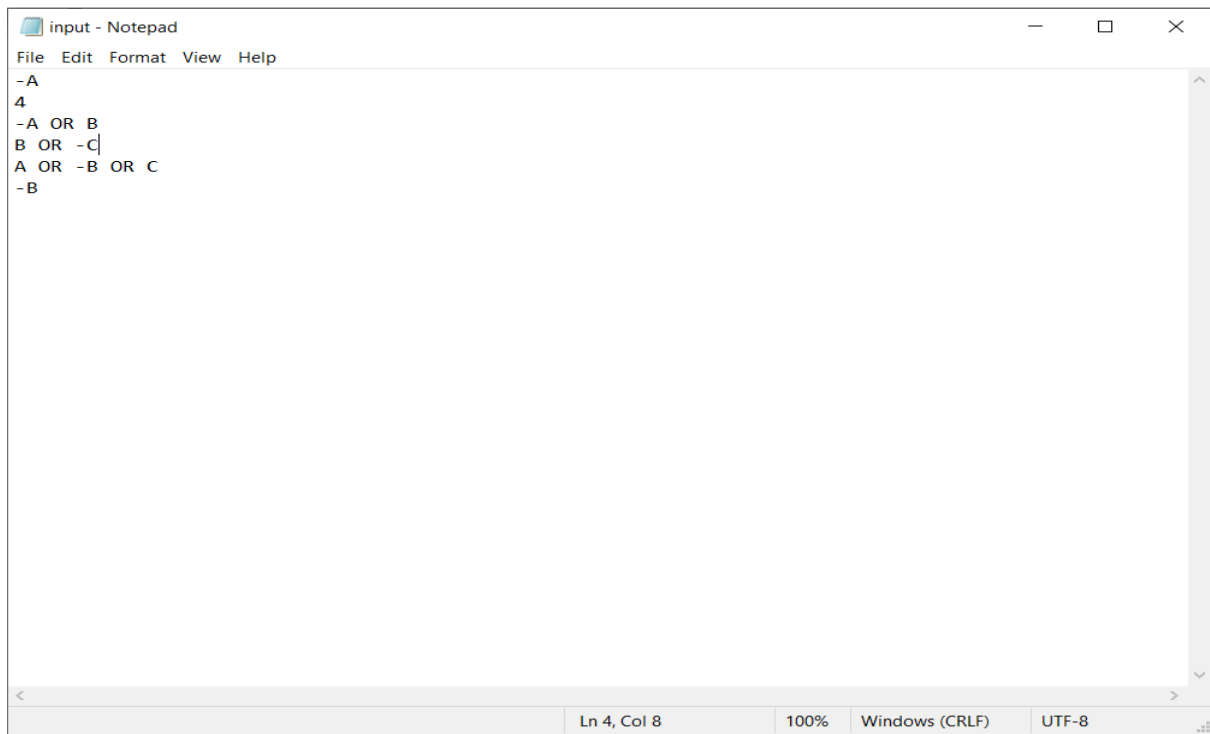
#### 2.1.1. Đặc tả dữ liệu đầu vào và đầu ra

- Các file input và output có định dạng như yêu cầu bài toán.
- Giữa các literal và từ khóa cách nhau bởi một khoảng trắng. Ví dụ: A OR B, -B OR C, ...
- Câu alpha chỉ gồm từ đơn (ví dụ: A, B, -C,...)
- Các literal trong cùng mệnh đề được sắp xếp theo thứ tự chữ cái: Chưa thực hiện được.

#### 2.1.2. Báo cáo test case

##### Test case 1:

- Input:

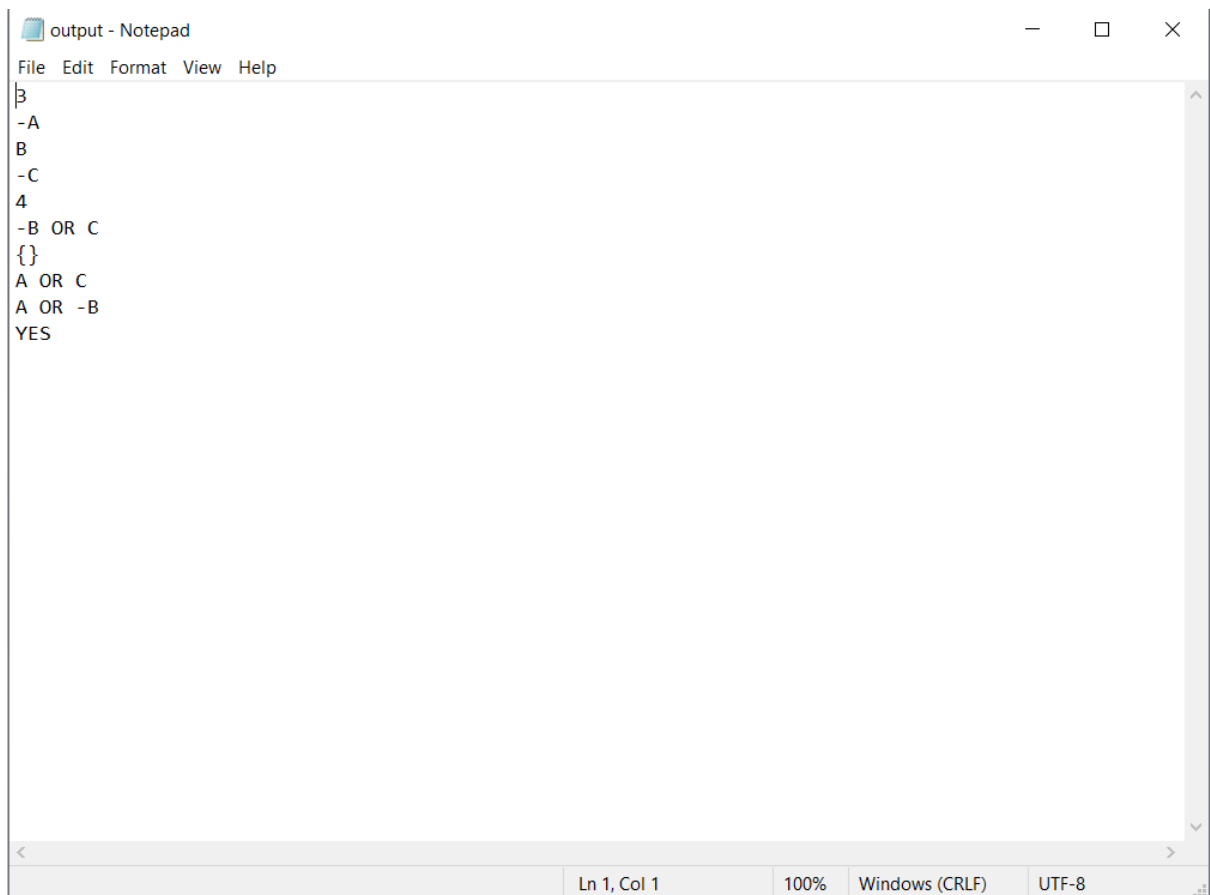


A screenshot of a Notepad window titled "input - Notepad". The window has a menu bar with "File", "Edit", "Format", "View", and "Help". The text content is as follows:

```
-A
4
-A OR B
B OR -C
A OR -B OR C
-B
```

The status bar at the bottom indicates "Ln 4, Col 8", "100%", "Windows (CRLF)", and "UTF-8".

- Output:



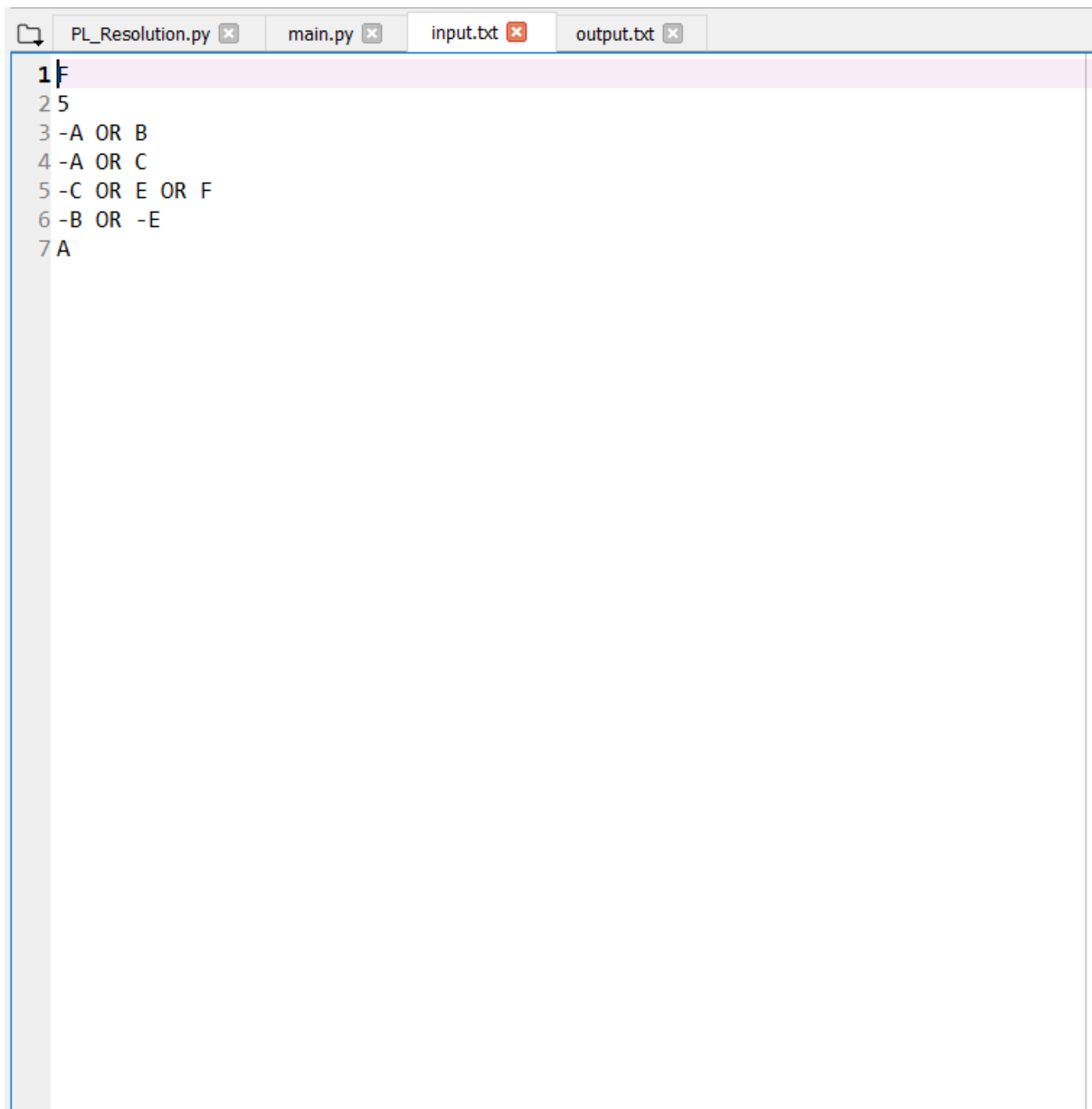
A screenshot of a Notepad window titled "output - Notepad". The window has a menu bar with "File", "Edit", "Format", "View", and "Help". The text content is as follows:

```
3
-A
B
-C
4
-B OR C
{}
A OR C
A OR -B
YES
```

The status bar at the bottom indicates "Ln 1, Col 1", "100%", "Windows (CRLF)", and "UTF-8".

## Test case 2:

- Input:



The screenshot shows a code editor with four tabs: 'PL\_Resolution.py', 'main.py', 'input.txt', and 'output.txt'. The 'input.txt' tab is active and contains the following text:

```
1 F
2 5
3 -A OR B
4 -A OR C
5 -C OR E OR F
6 -B OR -E
7 A
```

- Output:

```
PL_Resolution.py x main.py x input.txt x output.txt x
1 6
2 -A OR -E
3 B
4 -A OR E OR F
5 C
6 -C OR F OR -B
7 -C OR E
8 11
9 -A OR -C OR F
10 -E
11 -A OR F
12 -A OR -C
13 -C OR F
14 -A OR F OR -B
15 E OR F
16 -A OR E
17 F OR -B
18 E
19 -C OR -B
20 6
21 -C
22 F
23 -A
24 {}
25 -A OR -B
26 -B
27 YES
```

### Test case 3:

- Input:

```
PL_Resolution.py x main.py x input.txt x output.txt x
1 E
2 7
3 -A OR B
4 -A OR C OR E
5 -B OR -C OR D
6 -E OR F
7 -F OR G
8 -D OR G
9 A
```

- Output:

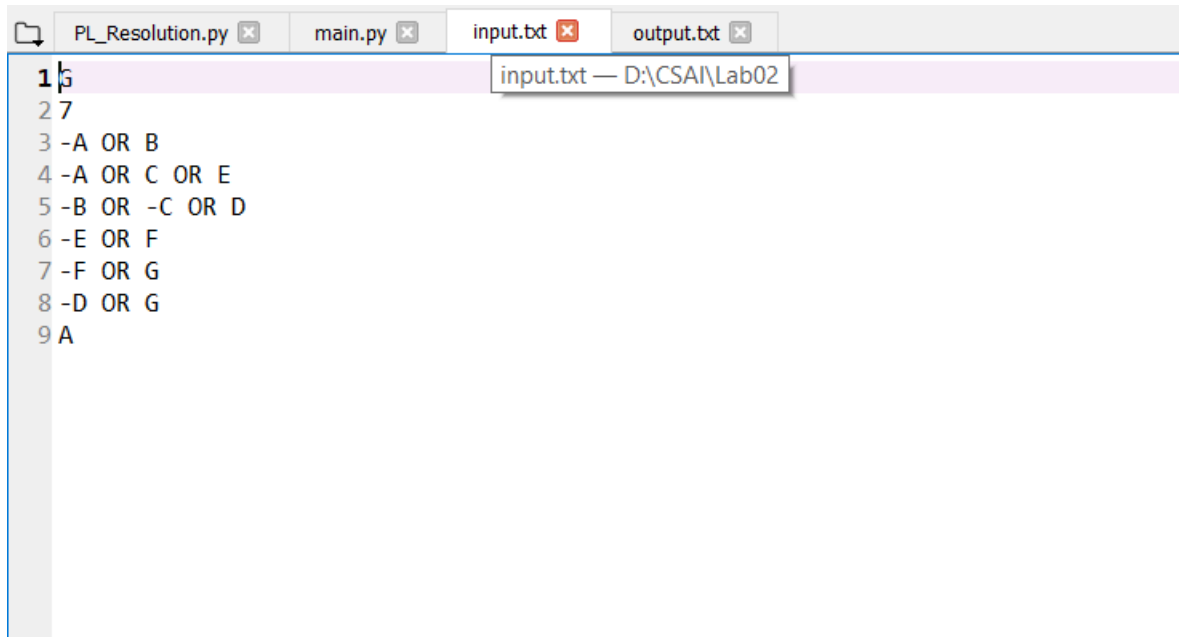
```

PL_Resolution.py x main.py x input.txt x output.txt x
1 8
2 -A OR -C OR D
3 B
4 -A OR E OR -B OR D
5 -A OR C OR F
6 C OR E
7 -A OR C
8 -B OR -C OR G
9 -E OR G
10 18
11 -A OR D OR E
12 -A OR -C OR G
13 -C OR D
14 -A OR D OR F
15 -A OR D
16 -C OR G
17 -A OR -B OR D OR F
18 -A OR E OR -B OR G
19 E OR -B OR D
20 -A OR -B OR D
21 -A OR -B OR D OR G
22 -A OR C OR G
23 C OR F
24 -A OR F OR -B OR G
25 C
26 E OR -B OR G
27 C OR G
28 -A OR -B OR G
29 16
30 -A OR E OR G
31 D OR E
32 -A OR D OR G
33 -A OR G OR F
34 -A OR G
35 D OR F
36 D
37 D OR G
38 G OR E
39 G OR F
40 G
41 -B OR D OR F
42 -B OR D
  -B OR D OR G
37 D OR G
38 G OR E
39 G OR F
40 G
41 -B OR D OR F
42 -B OR D
43 -B OR D OR G
44 F OR -B OR G
45 -B OR G
46 ∅
47 NO

```

#### Test case 4:

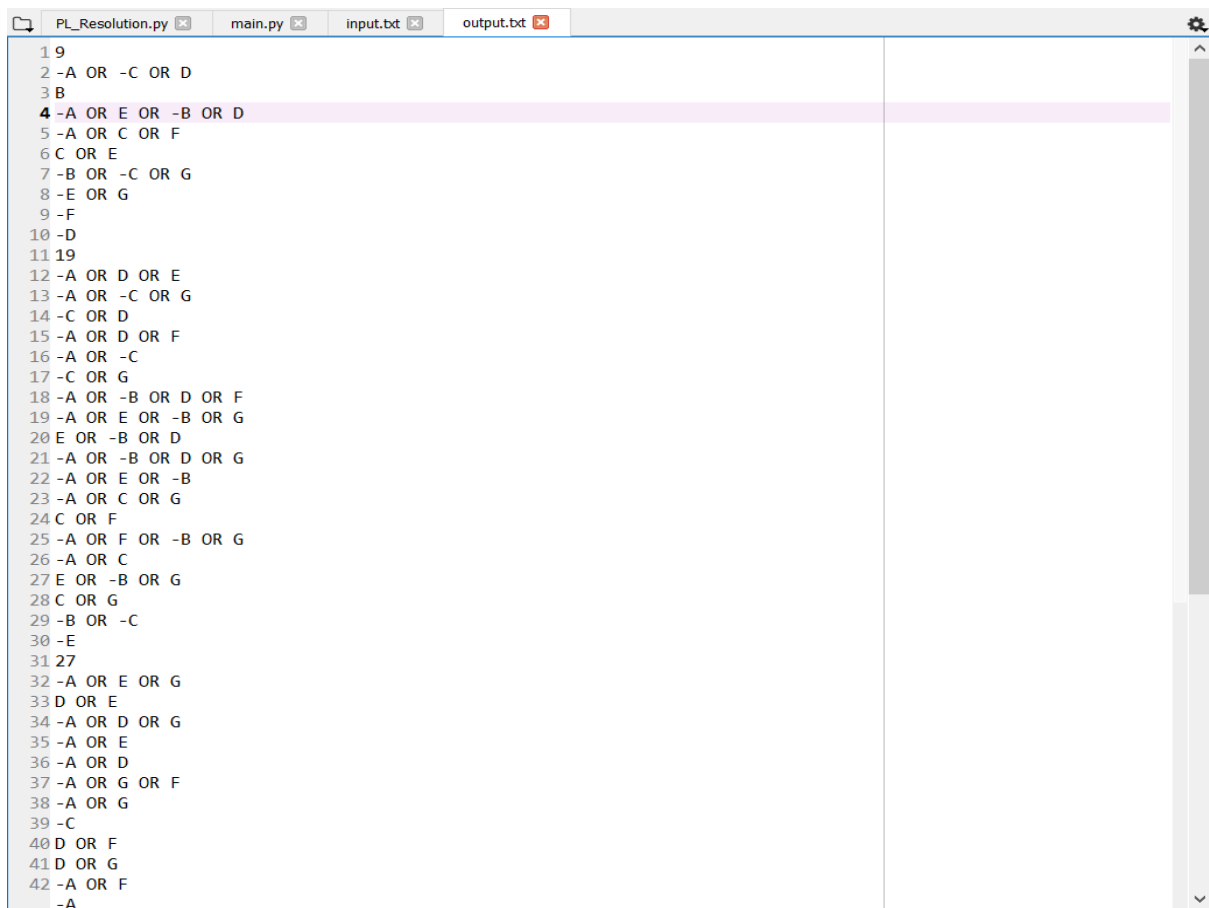
- Input:



The screenshot shows a text editor with four tabs: 'PL\_Resolution.py', 'main.py', 'input.txt', and 'output.txt'. The 'input.txt' tab is active, displaying the following text:

```
1 9
2 7
3 -A OR B
4 -A OR C OR E
5 -B OR -C OR D
6 -E OR F
7 -F OR G
8 -D OR G
9 A
```

- Output:



The screenshot shows the same text editor with the 'output.txt' tab active. It displays a list of 42 logical expressions, each preceded by a line number from 1 to 42. The expressions are:

```
1 9
2 -A OR -C OR D
3 B
4 -A OR E OR -B OR D
5 -A OR C OR F
6 C OR E
7 -B OR -C OR G
8 -E OR G
9 -F
10 -D
11 19
12 -A OR D OR E
13 -A OR -C OR G
14 -C OR D
15 -A OR D OR F
16 -A OR -C
17 -C OR G
18 -A OR -B OR D OR F
19 -A OR E OR -B OR G
20 E OR -B OR D
21 -A OR -B OR D OR G
22 -A OR E OR -B
23 -A OR C OR G
24 C OR F
25 -A OR F OR -B OR G
26 -A OR C
27 E OR -B OR G
28 C OR G
29 -B OR -C
30 -E
31 27
32 -A OR E OR G
33 D OR E
34 -A OR D OR G
35 -A OR E
36 -A OR D
37 -A OR G OR F
38 -A OR G
39 -C
40 D OR F
41 D OR G
42 -A OR F
-A
```

```
PL_Resolution.py x main.py x input.txt x output.txt x
24 C OR F
25 -A OR F OR -B OR G
26 -A OR C
27 E OR -B OR G
28 C OR G
29 -B OR -C
30 -E
31 27
32 -A OR E OR G
33 D OR E
34 -A OR D OR G
35 -A OR E
36 -A OR D
37 -A OR G OR F
38 -A OR G
39 -C
40 D OR F
41 D OR G
42 -A OR F
43 -A
44 G OR E
45 G OR F
46 G
47 -B OR D OR F
48 -A OR -B OR D
49 -A OR -B OR F
50 -A OR -B OR G
51 -B OR D OR G
52 E OR -B
53 -B OR D
54 -A OR -B
55 F OR -B OR G
56 C
57 F OR -B
58 -B OR G
59 5
60 E
61 D
62 F
63 {}
64 -B
65 YES
```

### Test case 5:

- Input:

```
PL_Resolution.py x main.py x input.txt x output.txt x
1 b
2 5
3 -A OR B OR D
4 -D OR E
5 -D OR F
6 -A OR -B OR -E
7 A
```



- Output:

```

PL_Resolution.py x main.py x input.txt x output.txt x
1 7
2 -A OR B OR E
3 -A OR B OR F
4 -A OR D OR -E
5 B OR D
6 -A OR B
7 -D OR -A OR -B
8 -B OR -E
9 10
10 B OR E
11 -A OR E OR -D
12 -A OR F OR -E
13 B OR F
14 -A OR F OR -D
15 D OR -E
16 -A OR -E
17 B
18 -A OR -D
19 -D OR -B
20 2
21 F OR -E
22 -E
23 0
24 NO

```

### Test case 6:

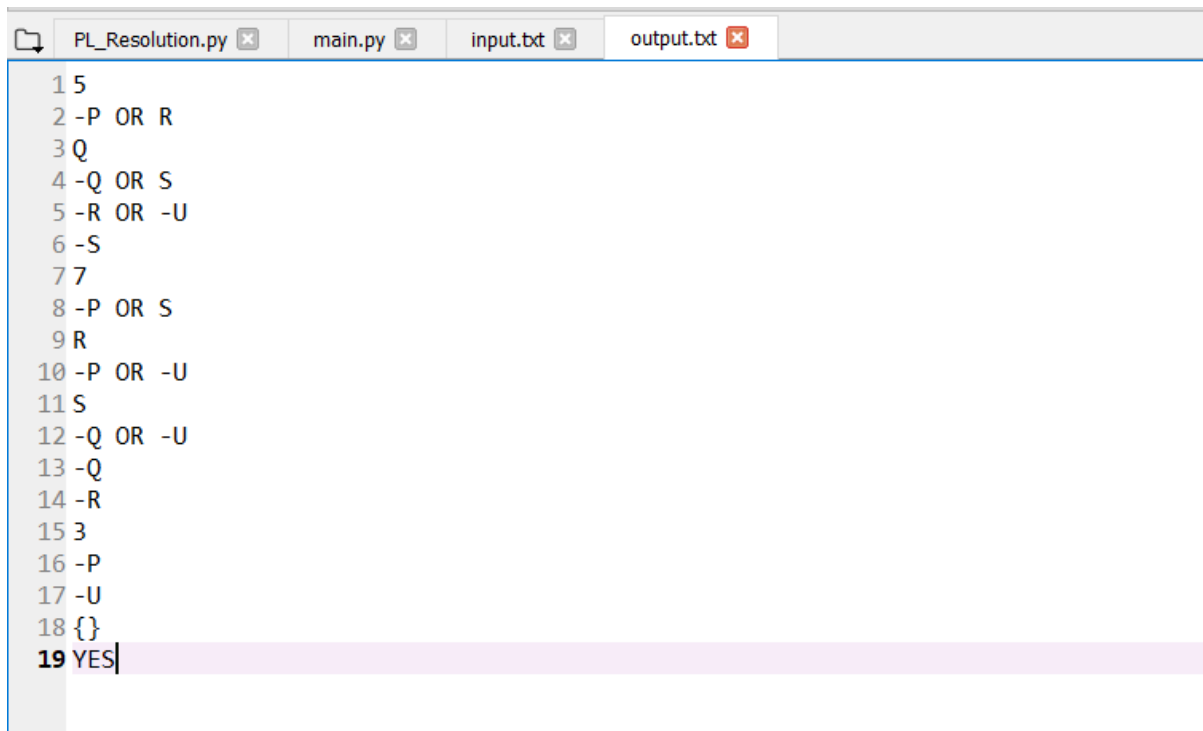
- Input:

```

PL_Resolution.py x main.py x input.txt x output.txt x
1 -U
2 5
3 -P OR Q
4 -Q OR R
5 -R OR S
6 -U OR -S
7 P

```

- Output:



```
PL_Resolution.py x main.py x input.txt x output.txt x
1 5
2 -P OR R
3 Q
4 -Q OR S
5 -R OR -U
6 -S
7 7
8 -P OR S
9 R
10 -P OR -U
11 S
12 -Q OR -U
13 -Q
14 -R
15 3
16 -P
17 -U
18 {}
19 YES
```

### 2.1.3. Đánh giá giải thuật hợp giải trên logic mệnh đề

- Ưu điểm:
  - Đơn giản, dễ cài đặt.
  - Luôn kiểm tra được câu alpha là đúng hay sai.
  - Thuật toán hợp giải có sức mạnh rất lớn vì nó có tính đầy đủ.
- Khuyết điểm:
  - Chi phí không gian lớn, tốn bộ nhớ nhiều, vì sau mỗi vòng lặp, kích thước KB tăng do có thêm các mệnh đề mới.
  - Thuật toán đòi hỏi vét cạn qua tất cả các trường hợp, nên thời gian thực hiện phụ thuộc vào thứ tự các bước thực hiện. Ví dụ, với một vài trường hợp, nếu là con người thực hiện thì chỉ cần hợp giải vài bước là có thể có được kết luận, nhưng máy phải vét cạn tất cả các trường hợp → không tối ưu.
  - Sau mỗi vòng lặp, có sự lặp lại việc hợp giải các mệnh đề mà ở vòng lặp trước đã hợp giải. Giải pháp là bỏ qua các mệnh đề cũ trong KB, chỉ xét các mệnh đề mới sinh ra và hợp giải chúng với các mệnh đề trong KB (sau khi đã thêm các mệnh đề mới này vào KB) (trong thuật toán cài đặt, đã sử dụng ý này).
- Đề xuất giải pháp:
  - Để tránh hợp giải lại các câu đã hợp giải, chỉ xét các mệnh đề mới sinh ra. Tương tự như trong hợp giải Robinson, ta có thể bỏ các mệnh đề cha của mệnh đề mới, chỉ giữ mệnh đề con được hợp giải. Như vậy thời gian xử lý ở vòng lặp sau có thể cải thiện, đồng thời chi phí không gian cũng giảm, vì càng ngày KB càng nhỏ lại.
  - Thuật toán hợp giải có sức mạnh đầy đủ, nhưng trong vài trường hợp là không cần thiết. Ta có thể sử dụng các thuật toán cải tiến của nó là suy diễn tiến hoặc lùi tùy thuộc vào các yêu cầu đặt ra:

- Suy diễn tiến là quá trình dựa trên dữ liệu, phù hợp với việc nhận dạng đối tượng, việc đưa ra quyết định, ...
- Suy diễn lùi là quá trình hướng tới mục tiêu, phù hợp với việc giải quyết vấn đề.

## 2.2. Biểu diễn tri thức bằng logic bậc nhất với Prolog

### 2.2.1. Xây dựng cây gia phả cho Hoàng gia Anh

Bằng ngôn ngữ lập trình Prolog (bắt nguồn từ “programming in logic”), xây dựng cây gia phả cho Hoàng gia Anh trên công cụ SWI-Prolog.

Một chương trình Prolog là tập hợp của *facts*, *rules* và *queries*. Để xây dựng hệ tri thức mô tả quan hệ trong Hoàng gia Anh phải dùng cú pháp của ngôn ngữ Prolog xây dựng *facts* và *rules* và dùng *queries* để kiểm tra, hỏi lại hệ tri thức vừa xây dựng.

- Xây dựng *facts* bằng các vị từ ban đầu theo yêu cầu: *parent*(Parent,Child), *male*(Person), *married*(Person,Person), *female*(Person), *divorced*(Person,Person).

```
male(prince_andrew).
male(prince_william).
male(prince_harry).
male(prince_george).
male(prince_louis).
male(archie_harrison).
male(mark_phillips).
male(timothy_laurence).
male(prince_edward).
male(peter_phillips).
male(mike_tindall).
male(james).
```

*Fact của vị từ male(Person) theo cây phả hệ*

- Định nghĩa thêm các vị từ mới từ các vị từ ban đầu bằng *rules*:
  - Quan hệ *married*(Person1, Person2) cần thêm rules để xử lý tính đối xứng của quan hệ này.

```
%rule
married(Person1,Person2) :- married(Person2,Person3), Person1 = Person3, !.
divorced(Person1,Person2) :- divorced(Person2,Person3), Person1 = Person3, !.
```

*Rule xử lý tính đối xứng của quan hệ married và divorced*

- Quan hệ *sibling* (Person1, Person2) – quan hệ đối xứng, khi *query* bị kết quả trùng lặp.

```
:- table sibling/2.
sibling(Person1,Person2):-
    parent(X,Person1),
    parent(X,Person2),
    Person1\=Person2.
```

- Xây dựng *queries* để kiểm tra hệ tri thức vừa xây dựng:

```

mother(X,prince_andrew).
wife(queen_elizabeth,mike_tindall).
son(X,prince_william).
grandparent(X,prince_beatrice).
sibling(X,prince_charlotte).
brother(X,lady_louise_windsor).
uncle(X,prince_louis).
wife(diana,prince_charles).
daughter(X,prince_phillip).
aunt(X,james).
niece(X,prince_harry).
sister(X,prince_louis).
nephew(X,princess_anne).
grandmother(X,peter_phillips).
grandmother(X,princess_beatrice).
husband(X,princess_anne).
uncle(X,princess_beatrice).
mother(camilla_parker_bowles,prince_william).
sister(princess_charlotte,archie_harrison).
parent(X,prince_harry).
husband(mark_phillips,princess_anne).
grandfather(X,Y).

```

## Bộ 20 câu hỏi cây phả hệ chồ gia đình Hoàng gia Anh:

- **Q1: Ai là mẹ của Prince Andrew?**

```

?- mother(X,prince_andrew).
X = queen_elizabeth ;
false.

```

- **Q2: Nữ hoàng Elizabeth có phải là vợ của Mia Grace Tindall không?**

```

?- wife(queen_elizabeth,mike_tindall).
false.

```

- **Q3: Con trai của Prince William?**

```

?- son(X,prince_william).
X = prince_george ;
X = prince_louis.

```

- **Q4: Ông bà của Princess Beatrice là ai?**

```

?- grandparent(X,princess_beatrice).
X = queen_elizabeth ;
X = prince_philip ;
false.

```

- **Q5: Anh em của Princess Charlotte là ai?**

```
?- sibling(X,princess_charlotte).  
X = prince_george ;  
X = prince_louis.
```

- Anh em trai của Lady Louise Windsor là gì?

```
?- brother(X,lady_louise_windsor).  
X = james ;  
false.
```

- Q6: Chú của Prince Louis là ai?

```
?- uncle(X,prince_louis).  
X = prince_harry.
```

- Q7: Diana hiện tại có phải là vợ của Prince Charles không?

```
?- wife(diana,prince_charles).  
false.
```

- Q8: Con gái của Prince Phillip là ai?

```
?- daughter(X,prince_philip).  
X = princess_anne ;  
false.
```

- Q9: Cô của James là ai?

```
?- aunt(X, james).  
X = princess_anne ;  
X = camilla_parker_bowles ;  
X = sarah_ferguson ;  
false.
```

- Q10: Cháu gái (niece) của Prince Harry là ai?

```
?- niece(X,prince_harry).  
X = princess_charlotte ;  
false.
```

- Q11: Chị (em) gái của Prince Louis là ai?

```
?- sister(X,prince_louis).  
X = princess_charlotte ;  
false.
```

- **Q12: Cháu trai (nephew) của Princess Anne là ai?**

```
?- nephew(X,princess_anne).  
X = prince_william ;  
X = prince_harry ;  
X = james.
```

- **Q13: Bà của Peter Phillips là ai?**

```
?- grandmother(X,peter_phillips).  
X = queen_elizabeth ;  
false.
```

- **Q14: Ông của Princess Beatrice là ai?**

```
?- grandfather(X,princess_beatrice).  
X = prince_philip ;  
false.
```

- **Q15: Chồng hiện tại của Princess Anne là ai?**

```
[1] ?- husband(X,princess_anne).  
X = timothy_laurence ;  
false.
```

- **Q16: Cậu của Princess Beatrice là ai?**

```
?- uncle(X, princess_beatrice).  
X = prince_charles ;  
X = prince_edward ;  
X = timothy_laurence ;  
false.
```

- **Q17: Camilla Parker Bowles có là mẹ của Prince William không?**

```
[1] ?- mother(camilla_parker_bowles,prince_william).  
false.
```

- **Q18: Princess Charlotte có là chị em với Archir Harrison không?**

```
[1] ?- sister(princess_charlotte,archie_harrison).  
false.
```

- **Q19: Bố mẹ của Prince Harry là ai ?**

```
[1] ?- parent(X,prince_harry).  
X = diana ;  
X = prince_charles.
```

- **Q20: Mark Phillips hiện tại có là chồng của Princess Anne không?**

```
[1] ?- husband(mark_phillips,princess_anne).  
false.
```

- **Q21: Liệt kê tất cả các cặp ông cháu?**

```
[1] ?- grandfather(X,Y).  
X = prince_philip,  
Y = prince_william ;  
X = prince_philip,  
Y = prince_harry ;  
X = prince_philip,  
Y = princess_eugenie ;  
X = prince_philip,  
Y = princess_beatrice ;  
X = prince_philip,  
Y = peter_phillips ;  
X = prince_philip,  
Y = zara_tindall ;  
X = prince_philip,  
Y = lady_louise_windsor ;  
X = prince_philip,  
Y = james ;  
X = prince_charles,  
Y = prince_george ;  
X = prince_charles,  
Y = princess_charlotte ;  
X = prince_charles,  
Y = prince_louis ;  
X = prince_charles,  
Y = archie_harrison ;  
false.
```