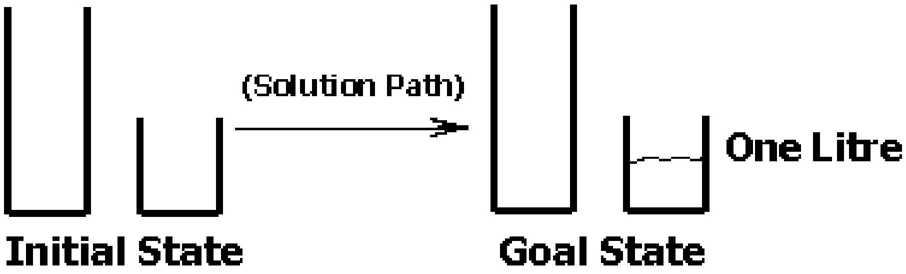
Experiment No 2

Aim: To study and implement Uninformed search method for Water Jug problem.

Theory:
We have one 3 litre jug, one 5 litre jug and an unlimited supply of water. The goal is to get exactly one litre of water into either jug. Either jug can be emptied or filled, or poured into the other.



A state in this problem could be represented with just a pair of numbers. The first representing the number of litres of water in the 5 litre (large) jug, and the second representing the number of litres of water in the 3 litre (small) jug.

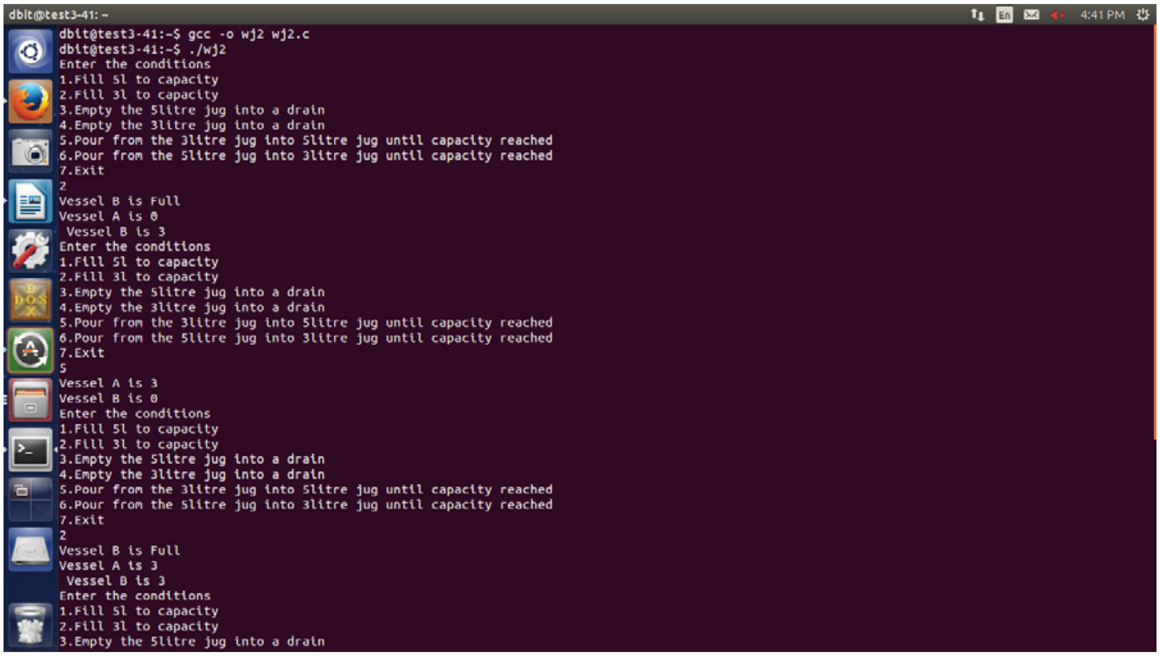
The initial state would typically be: (0, 0) to represent the fact that both jugs start off empty.
The final state would similarly be represented as: (0, 1).
The operators for this problem could include:

1. Fill the 5 litre jug to capacity from a water source
2. Fill the 3 litre jug to capacity from a water source
3. Empty the 5 litre jug into a drain
4. Empty the 3 litre jug into a drain
5. Pour from the 3 litre jug into 5 litre jug until capacity reached
6. Pour from the 5 litre jug into 3 litre jug until capacity reached

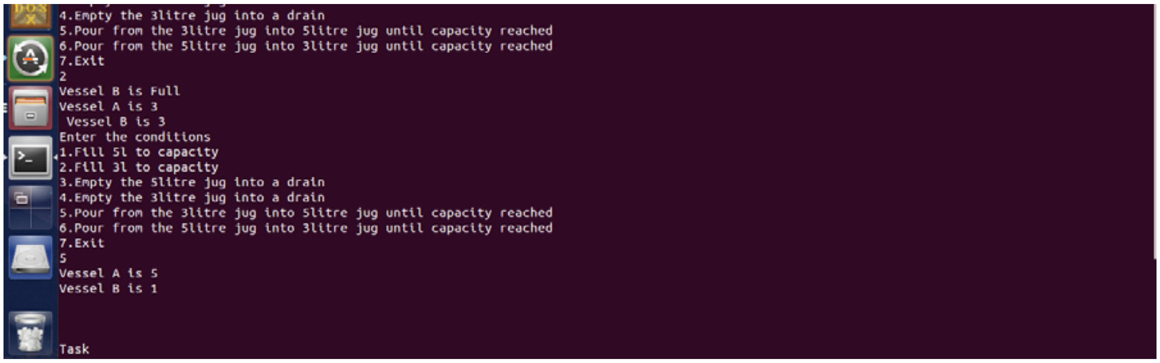
Code:

```
#include<stdio.h>
int main()
{
int a=0,b=0,r,temp1=0,temp2=0;
do
{
printf("Enter the conditions \n");
printf("1.Fill 5l to capacity \n");
printf("2.Fill 3l to capacity \n");
printf("3.Empty the 5litre jug into a drain\n");
printf("4.Empty the 3litre jug into a drain \n");
printf("5.Pour from the 3litre jug into 5litre jug until capacity reached\n");
printf("6.Pour from the 5litre jug into 3litre jug until capacity reached\n");
printf("7.Exit\n");
scanf("%d",&r);
switch(r)
{
case 1:
a=5; printf("Vessel A is full\n");
printf("Vessel A is %d \nVessel B is %d\n",a,b);
break;
case 2:
b=3; printf("Vessel B is Full\n");
printf("Vessel A is %d \n Vessel B is %d \n",a,b);
break;
case 3:
a=0; printf("Vessel A is Empty\n");
printf("Vessel A is %d \nVessel B is %d\n",a,b);
break;
case 4:
b=0;printf("Vessel B is Empty\n");
printf("Vessel A is %d \nVessel B is %d\n",a,b);
break;
case 5:
temp1=a+b;
if (temp1>5){
b=temp1-5;
a=5;
}
else
{
a=a+b;
b=0;
}
printf("Vessel A is %d\nVessel B is %d\n",a,b);
if((a==1)||(b==1))
printf("\n\n\nTask\n\n");
break;
case 6:
temp2=b+a;
if (temp2>3)
{
a=temp2 - 3;
b=3;
}
else
{
temp2=a+b;
b=temp2;
a=0;
}
printf("Vessel A is %d \nVessel B is %d\n",a,b);
if((a==1)||(b==1))
printf("\n\n\nTask Completed\n\n");
break;
case 7:
return 0;
}
}while(r!=7);
}
```

Output:



```
dbit@test3-41:~$ gcc -o wj2 wj2.c
dbit@test3-41:~$ ./wj2
Enter the conditions
1.Fill 5l to capacity
2.Fill 3l to capacity
3.Empty the 5litre jug into a drain
4.Empty the 3litre jug into a drain
5.Pour from the 3litre jug into 5litre jug until capacity reached
6.Pour from the 5litre jug into 3litre jug until capacity reached
7.Exit
2
Vessel B is Full
Vessel A is 0
 Vessel B is 3
Enter the conditions
1.Fill 5l to capacity
2.Fill 3l to capacity
3.Empty the 5litre jug into a drain
4.Empty the 3litre jug into a drain
5.Pour from the 3litre jug into 5litre jug until capacity reached
6.Pour from the 5litre jug into 3litre jug until capacity reached
7.Exit
5
Vessel A is 3
Vessel B is 0
Enter the conditions
1.Fill 5l to capacity
2.Fill 3l to capacity
3.Empty the 5litre jug into a drain
4.Empty the 3litre jug into a drain
5.Pour from the 3litre jug into 5litre jug until capacity reached
6.Pour from the 5litre jug into 3litre jug until capacity reached
7.Exit
2
Vessel B is Full
Vessel A is 3
 Vessel B is 3
Enter the conditions
1.Fill 5l to capacity
2.Fill 3l to capacity
3.Empty the 5litre jug into a drain
```



```
4.Empty the 3litre jug into a drain
5.Pour from the 3litre jug into 5litre jug until capacity reached
6.Pour from the 5litre jug into 3litre jug until capacity reached
7.Exit
2
Vessel B is Full
Vessel A is 3
 Vessel B is 3
Enter the conditions
1.Fill 5l to capacity
2.Fill 3l to capacity
3.Empty the 5litre jug into a drain
4.Empty the 3litre jug into a drain
5.Pour from the 3litre jug into 5litre jug until capacity reached
6.Pour from the 5litre jug into 3litre jug until capacity reached
7.Exit
5
Vessel A is 5
Vessel B is 1


Task
```

Conclusion:
An important aspect of intelligence is goal-based problem solving. The solution of many problems can be described by finding a sequence of actions that lead to a desirable goal. One such example is the Water Jug problem where we find a sequence of consecutive steps to rreach the end result.