

## Cross Word Puzzle (Backtracking – Uninformed Search)

### Summary:

It is difficult to fill the cells in this puzzle with the given input, imagine that you started to fill the table with the first word [zombifies] in H1 & [akecabele] in H2 & [brickwork] in H3 & [backcheck] in H4 after that you will find that V1 & V2 & V3 & V4 can't be filled with any of the remaining words. The strategy is to **backtrack** and try **another path** to fill words in all cells in (H & V) and return all valid solutions. If not reached to the solution, then it prints No solution.

This tricky game is solved here using the **Uninformed Search Technique**.

The program is written in Prolog.

	V1	V2	V3	V4
H1				
H2				
H3				
H4				

```
word(zombifies, [z.o.m.b.i.f.i.e.s]).
word(akecabele, [a.k.e.c.a.b.e.l.e]).
word(brickwork, [b.r.i.c.k.w.o.r.k]).
word(backcheck, [b.a.c.k.c.h.e.c.k]).
word(acmrremad, [a.c.m.r.r.e.m.a.d]).
word(nhgwpfabz, [n.h.g.w.p.f.a.b.z]).
word(jellybean, [j.e.l.l.y.b.e.a.n]).
word(aerreoded, [a.e.r.r.e.o.d.e.d]).
```

1- Input

8 words:

-----

word([z,o,m,b,i,f,i,e,s]).

word([a,k,e,c,a,b,e,l,e]).

word([b,r,i,c,k,w,o,r,k]).

word([b,a,c,k,c,h,e,c,k]).

word([a,c,m,r,r,e,m,a,d]).

word([n,h,g,w,p,f,a,b,z]).

word([j,e,l,l,y,b,e,a,n]).

word([e,a,r,r,e,o,d,e,d]).

\_\_\_\_\_

2-Output:

		V1		V2		V3		V4	
		e		a		n		a	
H1	b	a	c	k	c	h	e	c	k
		r		e		g		m	
H2	b	r	i	c	k	w	o	r	k
		e		a		p		r	
H3	z	o	m	b	i	f	i	e	s
		d		e		a		m	
H4	j	e	l	l	y	b	e	a	n
		d		e		z		d	