PreLab Part 1

1.Int array[26][26];
for (i=0;i<26;i++) {
	for (j=0;j<26;j++) {
		array[i][j]=0
}
OpenFile
fScanf(input,char):
char[0]-'A'
char[1]-'B'
Array[char[0]][char[1]]]

2. A->B->C->F->Z
A->B->C->Z
A->B->D->E FAIL

3. 1->2->4->5->4->2->1

PRELAB PART2

Stack_create :
stack(malloc(sizeof(stack))

Stack_delete:
free(stack)

Stack_empty:
Return s->top=0

Stack_size:
Return s->top

Stack_push:
If Stack == capacity:
Allocate more

s->items[s->top]=item
s->top+=1

Stack_pop:
s->top-=1
item=s->items[s->top]

Stack_print:
item+'A'
s->top=0
while(s->items[s->top] !=0:
Print
top++

PSEUDOCODE:
#include stack.c and stack .h
Int main {
Array[26][26]
For i =0;i<26 i++ {
	For j=0;j<26;j++ {
		array[i][j]=0
Create Stack
while(argv) !+EOF {
Case i:
fopen(file)
Fscanf (input)
input[0]-'A' and also input1

Array[input0][input1]

Case: d
Read input

stack_push(0)
Recursion()

Case u:
Recursion

Case m:
Print array
If (case u):	print [i][j] and [j][i]

RECURSION(int y):

If y==25 {
	stack_push(25)
	Stack_print
	Get # paths and shortest length
	Stack_pop

Return:

For possible:
    If ==1 and not visited:
        Stack_push(current)
        visited[i]=1
        RECURSION(next)
        POP
        visited=0:
    return