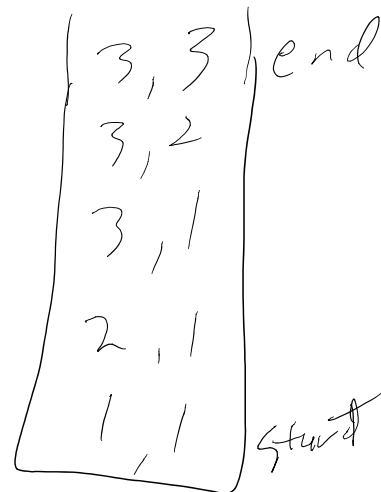


### Printing to Standard Error

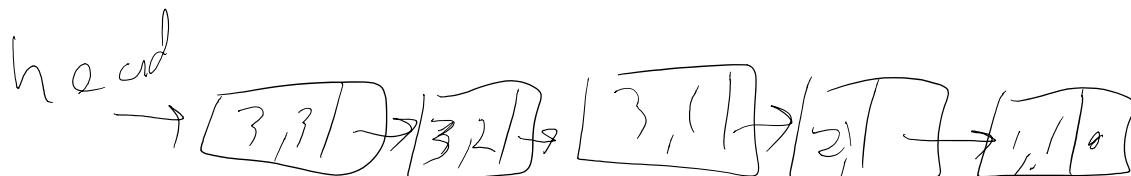
```

/* print to standard output */
printf ("Hello\n");
fprintf ( stdout, "Hello\n");

/* print to standard error */
fprintf (stderr, "Hello\n");
    
```



The output of the Path for Project 3.



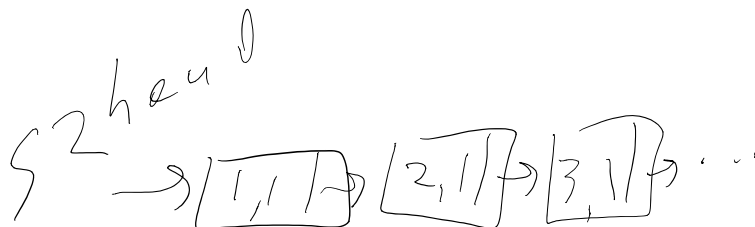
One solution is to use a second stack

- assume path solution is in s1.
- use a second stack s2.

```
reset (&s2);
```

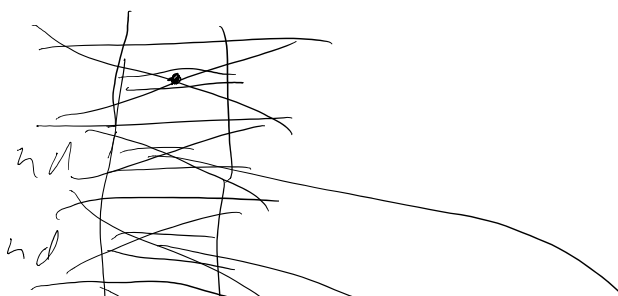
```

while ( isEmpty ( s1 ) == FALSE )
{
    push ( &s2, top (s1) );
    pop ( &s1 );
}
    
```



```

void show (node* hd)
{
    while (hd != NULL)
    {
        printf ("%d, ", hd->elem);
        hd = hd->next;
    }
}
    
```

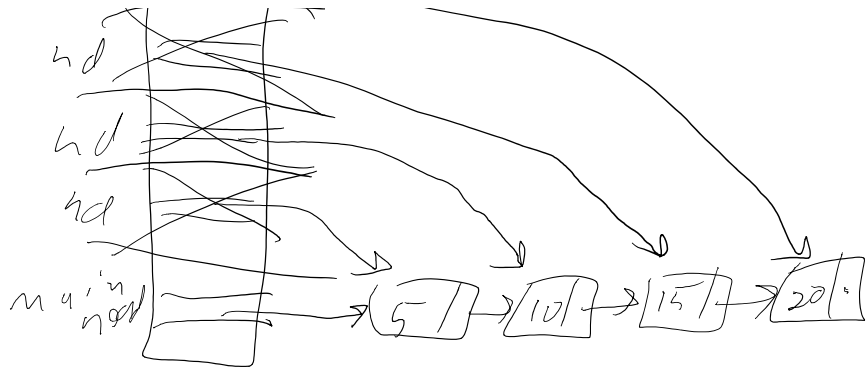


```

hd = hd->next;
}
}

void showR (node* hd)
{
if ( hd != NULL )
{
printf ("%d, ", hd->elem);
showR ( hd->next );
}
}

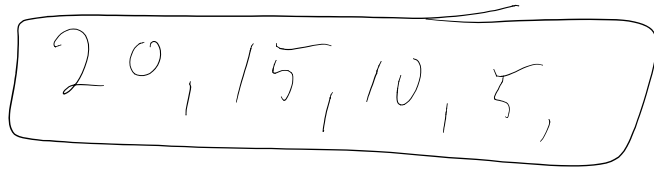
```



```

void showRReverse (node* hd)
{
if ( hd != NULL )
{
showRReverse ( hd->next );
printf ("%d, ", hd->elem);
}
}

```



For project 3, do NOT have "Memory Leaks"

==> Every malloc( ) should have a corresponding free( )