# Command Line Arguments

## Program #1: Echo, Version 1

This program echos its command line arguments on a separate line..

```
/*
 * print arguments followed by a newline
 */
#include <stdio.h>

int main(int argc, char *argv[])
{
      int i;                        /* counter in a for loop */

      /*
       * go through the argument list, skipping arg 0 (it's
       * the program name)
       */
      for(i = 1; i < argc-1; i++)
            printf("%s ", argv[i]);

      /*
       * now print the last argument, UNLESS it is arg 0
       */
      if (argc > 1)
            printf("%s\n", argv[argc-1]);

      /*
       * return success!
       */
      return(0);
}
```

## Program #2: Echo, Version 2

This also echos its command line arguments, but if the first argument is `-n`, it suppresses the trailing newline and does not print that argument.

```
#include <stdio.h>

int main(int argc, char *argv[])
{
      int prnl = 1;                 /* 1 to print newline, 0 not to */
      int i;                        /* counter in a for loop */

      /*
       * if argument 0 is -n, set prnl to suppress newline
       * skip it in the rinting; otherwise, print beginning with arg 1
       */
      if (argc > 1 &&
            argv[1][0] == '-' && argv[1][1] == 'n' && argv[1][2] == '\0'){
            prnl = 0;
            i = 2;
      }
      else
            i = 1;

      /*
       * go through the argument list, skipping arg 0 (it's
       * the program name) and arg 1 if needed
       */
      for( ; i < argc-1; i++)
      printf("%s ", argv[i]);

      /*
       * now print the last argument, UNLESS it is arg 0, or arg 1 and -n
       */
      if ((prnl && argc > 1) || (!prnl && argc > 2)){
            printf("%s", argv[argc-1]);
            /* put out the newline */
            if (prnl)
                  putchar('\n');
      }

      /*
       * return success!
       */
      return(0);
}
```