

## Parameters to main() - command line arguments

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    int i;

    for(i=1; i<argc; i++) {
        printf("[%s]", argv[i]);
    }
    printf("\n");
    return 0;
}
```

*Example:*

```
> gcc prog.c -o prog
>
> prog hi there
[hi][there]
>
> prog a      b c
[a][b][c]
>
```

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    int x, y;

    if(argc!=3) {
        fprintf(stderr, "usage err\n");
        exit(1);
    }
    x = atoi(argv[1]);
    y = atoi(argv[2]);
    printf("%d+%d=%d\n", x, y, x+y);
    return 0;
}
```

*Example:*

```
> gcc add.c -o add
>
> add 4 5
4+5=9
>
> add 4
usage err
>
```

## The 'copy' program example

```

#define CHUNK_LEN 100

int main(int argc, char *argv[])
{
    FILE *in = stdin;
    FILE *out = stdout;

    if(argc>3) error("usage err");
    if(argc>1) {
        in = fopen(argv[1], "r");
        if(in==NULL) error("Cannot read");
    }
    if(argc>2) {
        out = fopen(argv[2], "w");
        if(out==NULL) error("Cannot write");
    }

    copy(in, out);

    fclose(in);
    fclose(out);
    return 0;
}

void copy(FILE *source, FILE *dest)
{
    char buf[CHUNK_LEN];

    while(fgets(buf, CHUNK_LEN, source)) fputs(buf, dest);
}

void error(char *msg)
{
    fprintf(stderr, "Error: %s\n", msg);
    exit(1);
}

```

במידה והתוכנית מקבלת שני פרמטרים, מעתיקה את הקובץ אשר שמו הנו הפרמטר הראשון לקובץ אשר שמו הוא הפרמטר השני. אם קיבלה פרמטר אחד, מעתיקה את הקובץ אשר שמו הוא הפרמטר, לפלט הסטנדרטי. אם לא קיבלה פרמטרים כלל, מעתיקה את מה שמתקבל מהקלט הסטנדרטי לפלט הסטנדרטי.

*Example:*

```

> gcc copy.c -o copy

> copy filename
this is the content
> copy < filename
this is the content

```

## The difference in capturing stdout and stderr

```
#include <stdio.h>

int main()
{
    fprintf(stderr, "This is error\n");
    printf("This is stdout\n");

    return 0;
}
```

*Example:*

```
> gcc prog.c -o prog

> prog
This is error
This is stdout

> prog > out.txt
This is error
> cat out.txt
This is stdout

> prog >& errout.txt
> cat errout.txt
This is error
This is stdout

> (prog > out) >& err
> cat out
This is stdout
> cat err
This is error
```

## Memory corruption

```
#include <stdio.h>
#include <string.h>

int main()
{
    int n, *p;
    char str[8];

    n = 5;
    p = &n;
    *p = 10;

    printf("n = %d\n", n);

    strcpy(str, "Hi there");

    *p = 8;
    printf("n = %d\n", n);

    return 0;
}
```