```
#
# Makefile
#
# Computer Science 50
# Section 11
#


# compiler to use
CC = clang

# flags to pass compiler
CFLAGS = -ggdb3 -O0 -Qunused-arguments -std=c99 -Wall -Werror

# name for executable
EXE = mysum

# space-separated list of header files
HDRS =

# space-separated list of libraries, if any,
# each of which should be prefixed with -l
LIBS =

# space-separated list of source files
SRCS = mysum.c sumfunction.c

# automatically generated list of object files
OBJS = $(SRCS:.c=.o)


# default target
$(EXE): $(OBJS) $(HDRS) Makefile
    $(CC) $(CFLAGS) -o $@ $(OBJS) $(LIBS)

# dependencies
$(OBJS): $(HDRS) Makefile

# housekeeping
clean:
    rm -f core $(EXE) *.o
```

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

// use the sum function defined in sumfunction.c
int sum(int a, int b);

int main(int argc, char* argv[])
{
    // initialize variables to sum
    int a = 1;
    int b = 2;

    if (argc >= 2)
        a = strtol(argv[1], NULL, 10);
    if (argc >= 3)
        b = strtol(argv[2], NULL, 10);

    // print out the results of the sum function
    printf("%d + %d = %d\n", a, b, sum(a, b));
}
```

```
Here are some of the sum functions to try on your own:

int sum(int a, int b)
{
    return a + b;
}

unsigned sum(unsigned a, unsigned b)
{
    return a + b;
}

char *sum(char *a, int b)
{
    return &a[b];
}


int *sum(int *a, int b)
{
    return &a[b];
}

const unsigned char sum[] =
{
    0x8b, 0x44, 0x24, 0x08, 0x03, 0x44, 0x24, 0x04, 0xc3
};
```

```c
int sum(int a, int b) {
    return a + b;
}
```