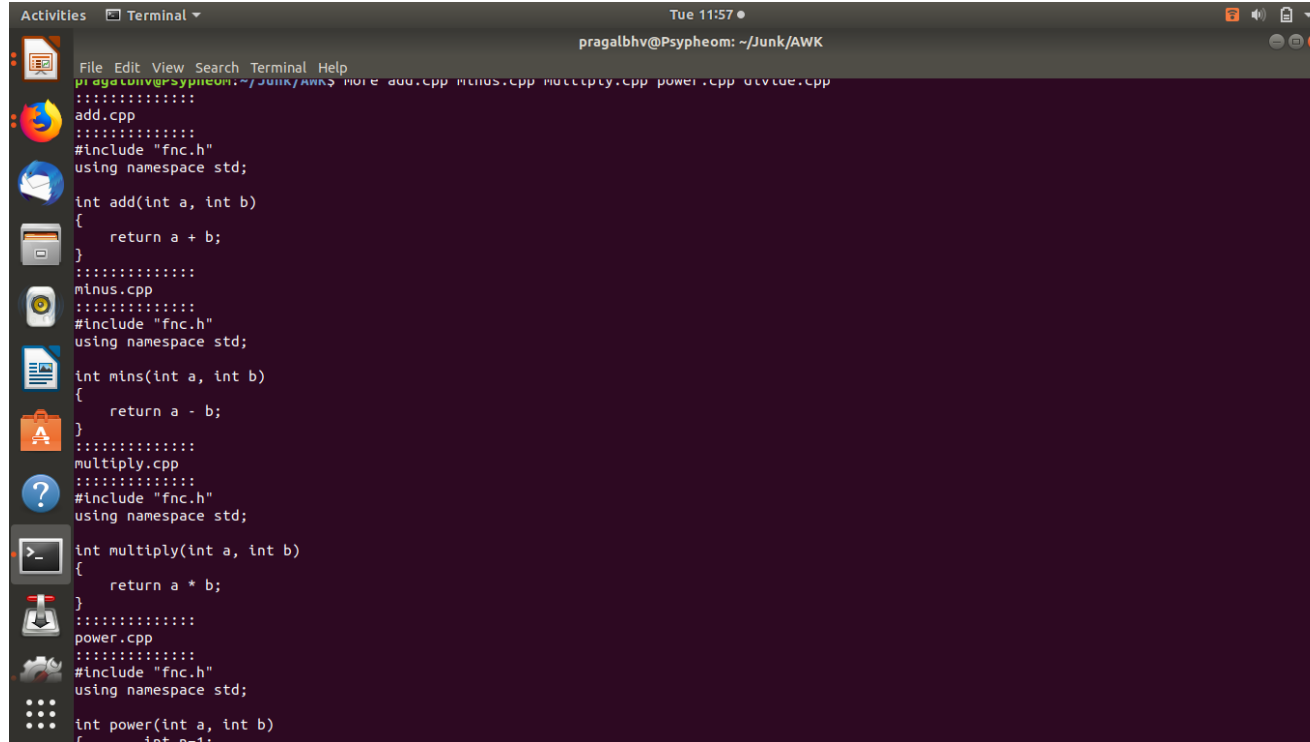


# HOMEWORK 5

Makefile

Take one of your old codes, split the code into separate files, one for each function. Create a makefile and test the recompilation



The screenshot shows a terminal window titled "pragalbhv@Pyspheim: ~/Junk/AWK" with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal displays the following code:

```
pragalbhv@Pyspheim:~/Junk/AWK$ more add.cpp minus.cpp multiply.cpp power.cpp divide.cpp
add.cpp
#include "fnc.h"
using namespace std;

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

minus.cpp
#include "fnc.h"
using namespace std;

int mins(int a, int b)
{
    return a - b;
}

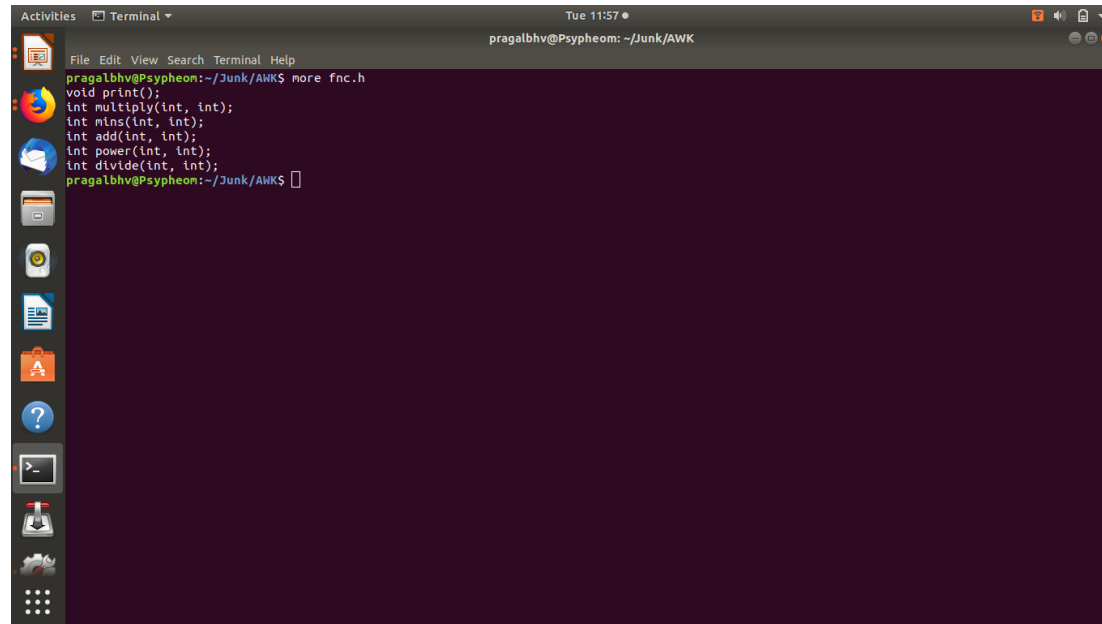
multiply.cpp
#include "fnc.h"
using namespace std;

int multiply(int a, int b)
{
    return a * b;
}

power.cpp
#include "fnc.h"
using namespace std;

int power(int a, int b)
{
    int n=1;
```

# Fnc.h



A terminal window titled "pragalbhv@Pspheon: ~/Junk/AWK" is shown. The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal displays the output of the command "more fnc.h", showing the following C code:

```
pragalbhv@Pspheon:~/Junk/AWK$ more fnc.h
void print();
int multiply(int, int);
int mins(int, int);
int add(int, int);
int power(int, int);
int divide(int, int);
pragalbhv@Pspheon:~/Junk/AWK$
```

The terminal window is part of a desktop environment with a sidebar on the left containing various application icons. The top of the window shows the system clock as "Tue 11:57".

```
Activities Terminal
pragalbhv@Pspheon:~/Junk/ANK$ more fnc.h
void print();
int multiply(int, int);
int mins(int, int);
int add(int, int);
int power(int, int);
int divide(int, int);
pragalbhv@Pspheon:~/Junk/ANK$
```

```
Activities Terminal Tue 12:00
pragalbhv@Pspheon:~/Junk/ANK$ make
g++ -g3 -ggdb -O main.o add.o minus.o multiply.o power.o divide.o -o main
pragalbhv@Pspheon:~/Junk/ANK$ ./main.e
ENTER TWO NOS.8
2
MULTIPLY16
MINUS6
ADD10
int DIVIDE4
POWER64
pragalbhv@Pspheon:~/Junk/ANK$
```

File Edit View Search Terminal Help

pragalbhv@Pyspheom:~/Junk/AWK\$ more Makefile

CC = g++

CC = g++

CFLAGS = -g3 -ggdb -O

default: main.o add.o minus.o divide.o power.o multiply.o function.h

\$(cc) \$(CFLAGS) main.o add.o minus.o multiply.o power.o divide.o -o main.e

main.o: main.cpp fnc.h

\$(cc) \$(CFLAGS) -c main.cpp -o main.o

add.o: add.cpp fnc.h

\$(cc) \$(CFLAGS) -c add.cpp -o add.o

minus.o: minus.cpp fnc.h

\$(cc) \$(CFLAGS) -c minus.cpp -o minus.o

multiply.o: multiply.cpp fnc.h

\$(cc) \$(CFLAGS) -c multiply.cpp -o multiply.o

divide.o: divide.cpp fnc.h

\$(cc) \$(CFLAGS) -c divide.cpp -o divide.o

power.o: power.cpp fnc.h

\$(cc) \$(CFLAGS) -c power.cpp -o power.o

DATESTAMP=\$(shell date +"%Y-%m-%d")

TARBALL=codebackup\_\$(DATESTAMP).tar

tar:

# ----- making a tarball -----

@echo "Backing up with datestamp: \$(DATESTAMP)";

@echo "Tarball name: \$(TARBALL)";

tar -cvf \$(TARBALL) \*.cpp \*.hpp readme.txt Makefile

@ls -l \$(TARBALL)

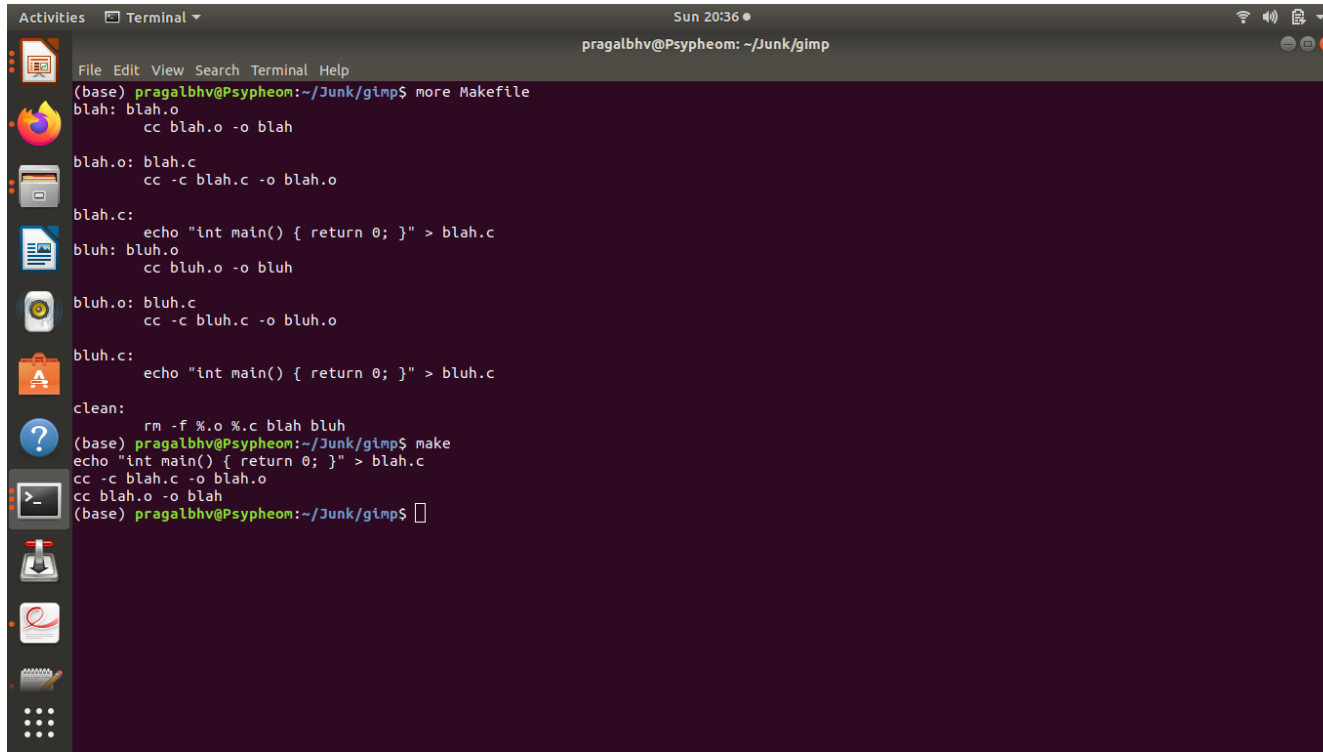
/bin/mv \$(TARBALL) \$(BACKUPDIR)

# ----- done moving tarball -----

# ----- end of Makefile -----

pragalbhv@Pyspheom:~/Junk/AWK\$

# Create a makefile that uses a pattern for files rather than explicit listing of each of the files.



```
pragalbhv@Psyspheom: ~/Junk/gimp
(base) pragalbhv@Psyspheom:~/Junk/gimp$ more Makefile
blah: blah.o
    cc blah.o -o blah

blah.o: blah.c
    cc -c blah.c -o blah.o

blah.c:
    echo "int main() { return 0; }" > blah.c
bluh: bluh.o
    cc bluh.o -o bluh

bluh.o: bluh.c
    cc -c bluh.c -o bluh.o

bluh.c:
    echo "int main() { return 0; }" > bluh.c

clean:
    rm -f %.o %.c blah bluh
(base) pragalbhv@Psyspheom:~/Junk/gimp$ make
echo "int main() { return 0; }" > blah.c
cc -c blah.c -o blah.o
cc blah.o -o blah
(base) pragalbhv@Psyspheom:~/Junk/gimp$
```