CS1217 - Spring 2023 - Homework 1

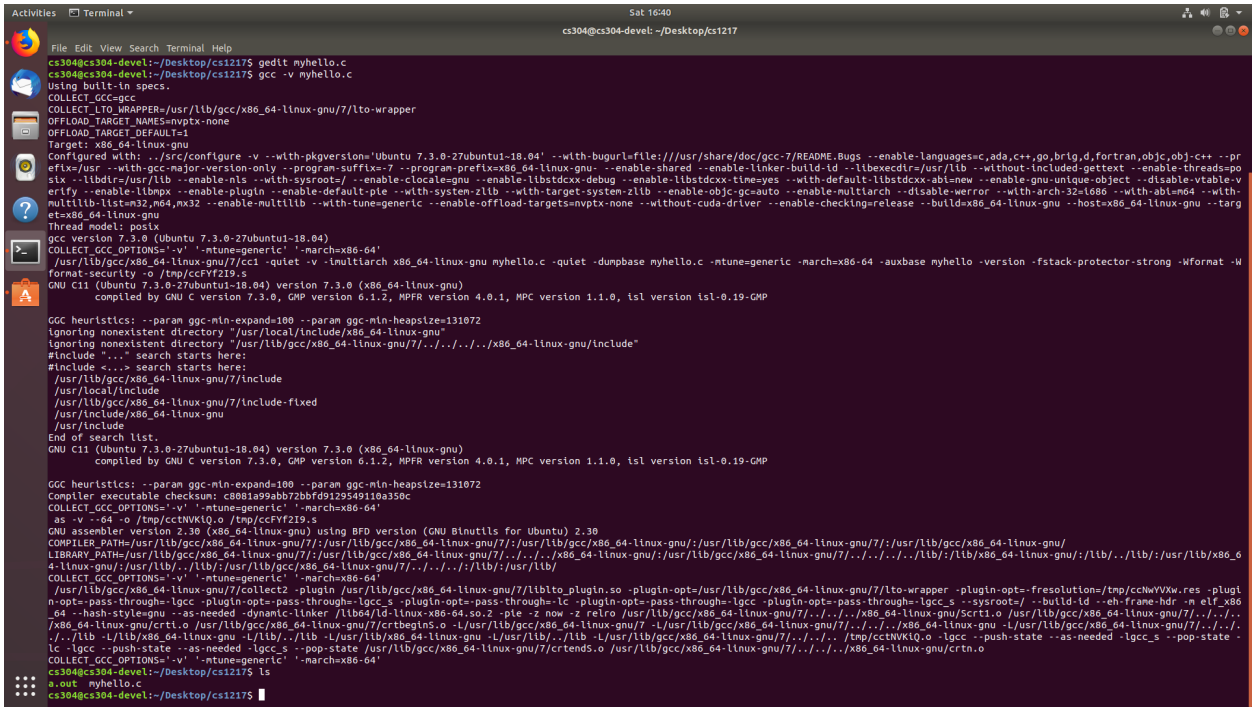
Bhumika Mittal, Saptarishi Dhanuka

Question 1

(a) The **#include** command is a compiler instruction that is used to include the contents of the file specified in the input stream during the compiling process. It tells the C preprocessor to look for the specified header file and include its content to the current code file before compiling the same.

(b) stdio stands for Standard Input and Output. stdio.h file is a built-in header file in C which is used to manage various functions related to input output process in C. This library felicitates the user communication with physical devices using streams.
#include <stdio.h> is specifically used to include the stdio.h header file to our current code base, which is mainly used to import the commands like printf, scanf, etc.

Question 2


Activities
Terminal
Sat 16:40
cs304@cs304-devel: ~/Desktop/cs1217
File Edit View Search Terminal Help
cs304@cs304-devel:~/Desktop/cs1217$ gedit myhello.c
cs304@cs304-devel:~/Desktop/cs1217$ gcc -v myhello.c
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/7/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 7.3.0-27ubuntu1~18.04' --with-bugurl=file:///usr/share/doc/gcc-7/README.Bugs --enable-languages=c,ada,c++,go,brig,d,fortran,objc,obj-c++ --prefix=/usr --with-gcc-major-version-only --program-suffix=-7 --program-prefix=x86_64-linux-gnu- --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --libdir=/usr/lib --enable-nls --with-sysroot=/ --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --enable-gnu-unique-object --disable-vtable-verify --enable-libmpx --enable-plugin --enable-default-pie --with-system-zlib --with-target-system-zlib --enable-objc-gc=auto --enable-multiarch --disable-werror --with-arch-32=i686 --with-abi=m64 --with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=generic --enable-offload-targets=nvptx-none --without-cuda-driver --enable-checking=release --build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=x86_64-linux-gnu
Thread model: posix
gcc version 7.3.0 (Ubuntu 7.3.0-27ubuntu1~18.04)
COLLECT_GCC_OPTIONS='-v' '-mtune=generic' '-march=x86-64'
/usr/lib/gcc/x86_64-linux-gnu/7/cc1 -quiet -v -imultiarch x86_64-linux-gnu myhello.c -quiet -dumpbase myhello.c -mtune=generic -march=x86-64 -auxbase myhello -version -fstack-protector-strong -Wformat -Wformat-security -o /tmp/ccFYf2I9.s
GNU C11 (Ubuntu 7.3.0-27ubuntu1~18.04) version 7.3.0 (x86_64-linux-gnu)
compiled by GNU C version 7.3.0, GMP version 6.1.2, MPFR version 4.0.1, MPC version 1.1.0, isl version isl-0.19-GMP
GGC heuristics: --param ggc-min-expand=100 --param ggc-min-heapsize=131072
ignoring nonexistent directory "/usr/local/include/x86_64-linux-gnu"
ignoring nonexistent directory "/usr/lib/gcc/x86_64-linux-gnu/7/../../../../x86_64-linux-gnu/include"
#include "..." search starts here:
#include <...> search starts here:
/usr/lib/gcc/x86_64-linux-gnu/7/include
/usr/local/include
/usr/lib/gcc/x86_64-linux-gnu/7/include-fixed
/usr/include/x86_64-linux-gnu
/usr/include
End of search list.
GNU C11 (Ubuntu 7.3.0-27ubuntu1~18.04) version 7.3.0 (x86_64-linux-gnu)
compiled by GNU C version 7.3.0, GMP version 6.1.2, MPFR version 4.0.1, MPC version 1.1.0, isl version isl-0.19-GMP
GGC heuristics: --param ggc-min-expand=100 --param ggc-min-heapsize=131072
Compiler executable checksum: c8081a99abb72bbfd9129549110a350c
COLLECT_GCC_OPTIONS='-v' '-mtune=generic' '-march=x86-64'
as -v --64 -o /tmp/cctNVklQ.o /tmp/ccFYf2I9.s
GNU assembler version 2.30 (x86_64-linux-gnu) using BFD version (GNU Binutils for Ubuntu) 2.30
COMPILER_PATH=/usr/lib/gcc/x86_64-linux-gnu/7/:/usr/lib/gcc/x86_64-linux-gnu/7/:/usr/lib/gcc/x86_64-linux-gnu/:/usr/lib/gcc/x86_64-linux-gnu/7/:/usr/lib/gcc/x86_64-linux-gnu/
LIBRARY_PATH=/usr/lib/gcc/x86_64-linux-gnu/7/:/usr/lib/gcc/x86_64-linux-gnu/7/../../../x86_64-linux-gnu/:/usr/lib/gcc/x86_64-linux-gnu/7/../../../../lib/:/lib/x86_64-linux-gnu/:/lib/../lib/:/usr/lib/x86_64-linux-gnu/:/usr/lib/../lib/:/usr/lib/gcc/x86_64-linux-gnu/7/../../../:/lib/:/usr/lib/
COLLECT_GCC_OPTIONS='-v' '-mtune=generic' '-march=x86-64'
/usr/lib/gcc/x86_64-linux-gnu/7/collect2 -plugin /usr/lib/gcc/x86_64-linux-gnu/7/liblto_plugin.so -plugin-opt=/usr/lib/gcc/x86_64-linux-gnu/7/lto-wrapper -plugin-opt=-fresolution=/tmp/ccNwYVXw.res -plugin-opt=-pass-through=-lgcc -plugin-opt=-pass-through=-lgcc_s -plugin-opt=-pass-through=-lc -plugin-opt=-pass-through=-lgcc -plugin-opt=-pass-through=-lgcc_s --sysroot=/ --build-id --eh-frame-hdr -m elf_x86_64 --hash-style=gnu --as-needed -dynamic-linker /lib64/ld-linux-x86-64.so.2 -pie -z now -z relro /usr/lib/gcc/x86_64-linux-gnu/7/../../../x86_64-linux-gnu/Scrt1.o /usr/lib/gcc/x86_64-linux-gnu/7/../../../x86_64-linux-gnu/crti.o /usr/lib/gcc/x86_64-linux-gnu/7/crtbeginS.o -L/usr/lib/gcc/x86_64-linux-gnu/7 -L/usr/lib/gcc/x86_64-linux-gnu/7/../../../x86_64-linux-gnu -L/usr/lib/gcc/x86_64-linux-gnu/7/../../../../lib -L/lib/x86_64-linux-gnu -L/lib/../lib -L/usr/lib/x86_64-linux-gnu -L/usr/lib/../lib -L/usr/lib/gcc/x86_64-linux-gnu/7/../../.. /tmp/cctNVklQ.o -lgcc --push-state --as-needed -lgcc_s --pop-state -lc -lgcc --push-state --as-needed -lgcc_s --pop-state /usr/lib/gcc/x86_64-linux-gnu/7/crtendS.o /usr/lib/gcc/x86_64-linux-gnu/7/../../../x86_64-linux-gnu/crtn.o
COLLECT_GCC_OPTIONS='-v' '-mtune=generic' '-march=x86-64'
cs304@cs304-devel:~/Desktop/cs1217$ ls
a.out myhello.c
cs304@cs304-devel:~/Desktop/cs1217$

Question 3

(a) "." in linux based systems generally denotes the current directory whereas "/" is a path separator. ./ is needed in front of a.out to locate the executable file (which is a.out). We need this because a.out is neither directly located in the path environment variable nor it is a built-in command.

(b) If we don't mention ./ in front of a.out, we get an error "command not found : a.out". This happens because the command is not in the path and is neither a built-in command. By saying ./a.out, we specify the path to the executable file and then execute it.

Question 4

(a) The following part of the Makefile is responsible for creating the myhello executable:

```
gcc -o myhello myhello.c
```

(b) Changing the name from what we created in step 3 (a.out) is a function of **gcc**. In line 4 of the Makefile, the gcc command compiles the myhello.c file and using the -o flag, we save the executable as myhello instead of the default a.out.

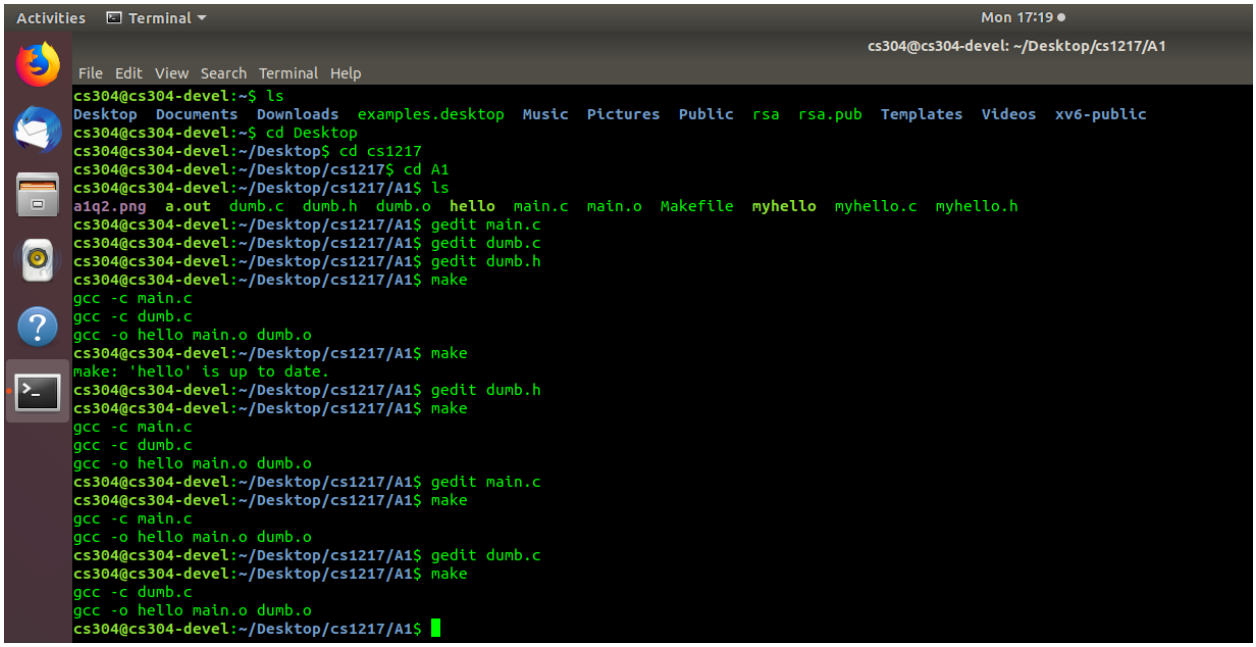
Question 5

(a) .c.o: is a double suffix rule[1] in the makefile which says take a .c file as a source suffix and compile it into an object file with .o as the target suffix. The -c flag is used to indicate the complier to compile the source code into an object code but not link it into an executable.

'$*.c' is an automatic variable which acts as the stem (the name of the file excluding the file extension/suffix) of the target filename. It is essentially a pattern matching tool which finds all the necessary .c files and copies their respective stems into their respective compiled .o files. So we get .o files with the same stem as their .c files

(b) This make does NOT recompile the binary if none of the source files have been changed, in this case. The attached screenshot depicts the same where it says 'hello' is up to date when none of the source files have been changed

(c) A change in any unrelated file does NOT forces the the recompilation of all the other files. Only the files which are related to the file in which the change is made recompiles. For example - the attached screenshot explains the same.



[1]Source: https://www.gnu.org/software/make/manual/make.html#Makefile-Contents

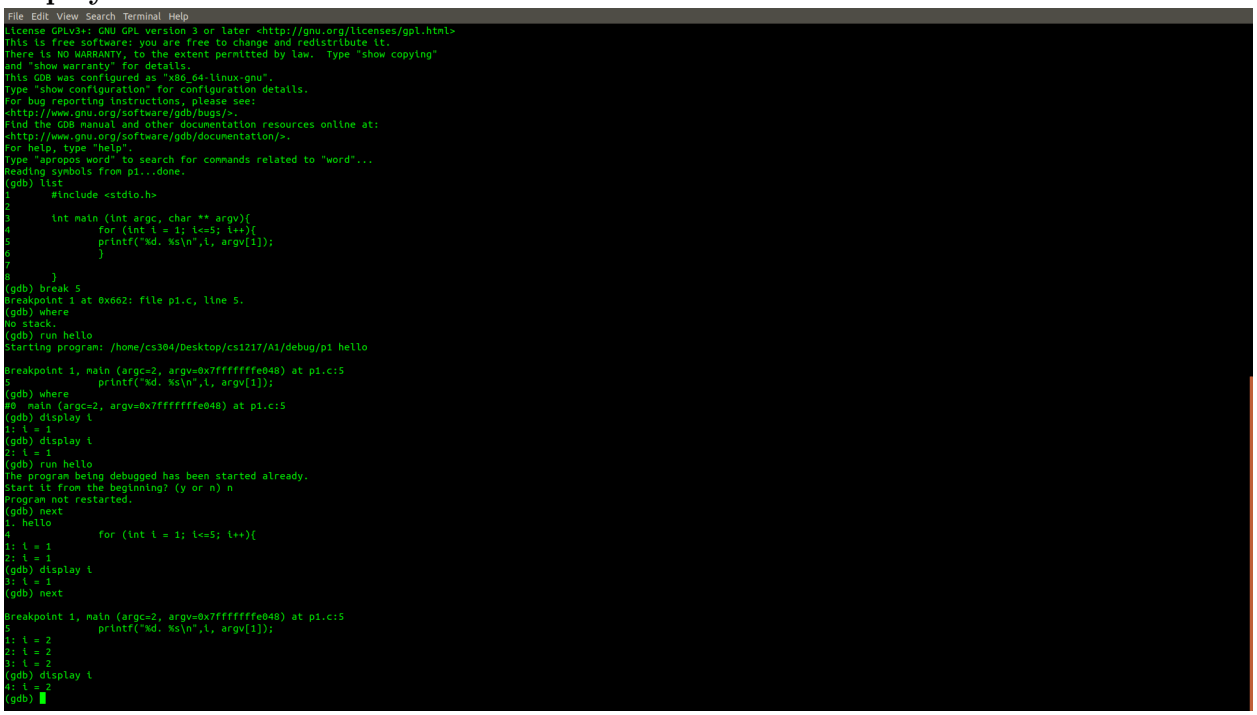
Question 6

The following screenshots show the output for the respective commands in gdb:

(a) **list**

```
File  Edit  View  Search  Terminal  Help
cs304@cs304-devel:~/Desktop/cs1217/A1/debug$ gdb p1
GNU gdb (Ubuntu 8.1.1-0ubuntu1) 8.1.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from p1...done.
(gdb) list
1       #include <stdio.h>
2
3       int main (int argc, char ** argv){
4               for (int i = 1; i<=5; i++){
5               printf("%d. %s\n",i, argv[1]);
6               }
7
8       }
(gdb)
```

(b) **display**


File Edit View Search Terminal Help
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from p1...done.
(gdb) list
1 #include <stdio.h>
2
3 int main (int argc, char ** argv){
4 for (int i = 1; i<=5; i++){
5 printf("%d. %s\n",i, argv[1]);
6 }
7
8 }
(gdb) break 5
Breakpoint 1 at 0x662: file p1.c, line 5.
(gdb) where
No stack.
(gdb) run hello
Starting program: /home/cs304/Desktop/cs1217/A1/debug/p1 hello
Breakpoint 1, main (argc=2, argv=0x7fffffffe048) at p1.c:5
5 printf("%d. %s\n",i, argv[1]);
(gdb) where
#0 main (argc=2, argv=0x7fffffffe048) at p1.c:5
(gdb) display i
1: i = 1
(gdb) display i
2: i = 1
(gdb) run hello
The program being debugged has been started already.
Start it from the beginning? (y or n) n
Program not restarted.
(gdb) next
1. hello
4 for (int i = 1; i<=5; i++){
1: i = 1
2: i = 1
(gdb) display i
3: i = 1
(gdb) next
Breakpoint 1, main (argc=2, argv=0x7fffffffe048) at p1.c:5
5 printf("%d. %s\n",i, argv[1]);
1: i = 2
2: i = 2
3: i = 2
(gdb) display i
4: i = 2
(gdb)

(c) **where**

```
File  Edit  View  Search  Terminal  Help
cs304@cs304-devel:~/Desktop/cs1217/A1/debug$ gdb p1
GNU gdb (Ubuntu 8.1.1-0ubuntu1) 8.1.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from p1...done.
(gdb) list
1       #include <stdio.h>
2
3       int main (int argc, char ** argv){
4               for (int i = 1; i<=5; i++){
5               printf("%d. %s\n",i, argv[1]);
6               }
7
8       }
(gdb) break 5
Breakpoint 1 at 0x662: file p1.c, line 5.
(gdb) run hello
Starting program: /home/cs304/Desktop/cs1217/A1/debug/p1 hello

Breakpoint 1, main (argc=2, argv=0x7fffffffe048) at p1.c:5
5               printf("%d. %s\n",i, argv[1]);
(gdb) where
#0  main (argc=2, argv=0x7fffffffe048) at p1.c:5
(gdb)
```

(d) **print**

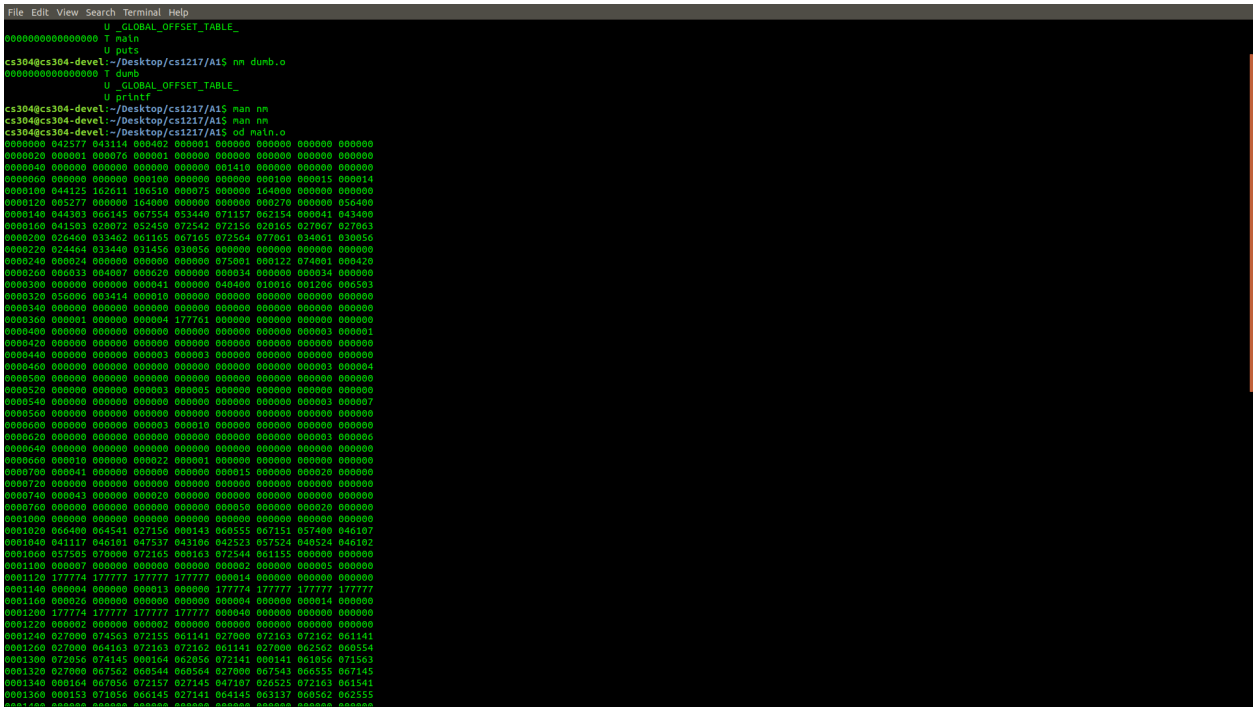
```
File  Edit  View  Search  Terminal  Help
cs304@cs304-devel:~/Desktop/cs1217/A1/debug$ gdb p1
GNU gdb (Ubuntu 8.1.1-0ubuntu1) 8.1.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from p1...done.
(gdb) list
1       #include <stdio.h>
2
3       int main (int argc, char ** argv){
4               for (int i = 1; i<=5; i++){
5               printf("%d. %s\n",i, argv[1]);
6               }
7
8       }
(gdb) break 5
Breakpoint 1 at 0x662: file p1.c, line 5.
(gdb) run hello
Starting program: /home/cs304/Desktop/cs1217/A1/debug/p1 hello

Breakpoint 1, main (argc=2, argv=0x7fffffffe048) at p1.c:5
5               printf("%d. %s\n",i, argv[1]);
(gdb) where
#0  main (argc=2, argv=0x7fffffffe048) at p1.c:5
(gdb) print argv[1]
$1 = 0x7fffffffe3b0 "hello"
(gdb)
```

Question 7

(a) **nm** - nm gives us information regarding symbols from the object file, executable or object-binary file. It displays the symbol value, symbol type (each letter means something different, eg - 'U' means the symbol is undefined.) and symbol name. If the file contains no symbol information, it displays that instead of giving any error. It can be used to differentiate between different types of files based on symbol type.

```
cs304@cs304-devel:~/Desktop/cs1217/A1/debug$ nm p1
0000000000201010 B __bss_start
0000000000201010 b completed.7698
                 w __cxa_finalize@@GLIBC_2.2.5
0000000000201000 D __data_start
0000000000201000 W data_start
0000000000000570 t deregister_tm_clones
0000000000000600 t __do_global_dtors_aux
0000000000200dc0 t __do_global_dtors_aux_fini_array_entry
0000000000201008 D __dso_handle
0000000000200dc8 d _DYNAMIC
0000000000201010 D _edata
0000000000201018 B _end
0000000000000714 T _fini
0000000000000640 t frame_dummy
0000000000200db8 t __frame_dummy_init_array_entry
000000000000086c r __FRAME_END__
0000000000200fb8 d _GLOBAL_OFFSET_TABLE_
                 w __gmon_start__
000000000000072c r __GNU_EH_FRAME_HDR
00000000000004f0 T _init
0000000000200dc0 t __init_array_end
0000000000200db8 t __init_array_start
0000000000000720 R _IO_stdin_used
                 w _ITM_deregisterTMCloneTable
                 w _ITM_registerTMCloneTable
0000000000000710 T __libc_csu_fini
00000000000006a0 T __libc_csu_init
                 U __libc_start_main@@GLIBC_2.2.5
000000000000064a T main
                 U printf@@GLIBC_2.2.5
00000000000005b0 t register_tm_clones
0000000000000540 T _start
0000000000201010 D __TMC_END__
cs304@cs304-devel:~/Desktop/cs1217/A1/debug$
```

(b) **od** - od is used to convert the contents of the argument file into different formats like octal, ASCII depending on the flag provided to it. Using this we can coherently see the contents of an object file or an executable file which would otherwise be human-unreadable. It writes the octal bytes (default, can be changed to other formats) of the file to the standard output.



(c) **objdump** - objdump can give us different kinds of detailed information about the given file. It takes an object file as argument along with atleast one flag (the information which needs to be displayed, refer attached screenshot for different flags). In the attached screenshot, we have used the -f flag to get information regarding the overall file header of the file. If the file given is not an object file, it gives an error about the same.

```
cs304@cs304-devel:~/Desktop/cs1217/A1$ objdump main.o
Usage: objdump <option(s)> <file(s)>
 Display information from object <file(s)>.
 At least one of the following switches must be given:
  -a, --archive-headers    Display archive header information
  -f, --file-headers       Display the contents of the overall file header
  -p, --private-headers    Display object format specific file header contents
  -P, --private=OPT,OPT... Display object format specific contents
  -h, --[section-]headers  Display the contents of the section headers
  -x, --all-headers        Display the contents of all headers
  -d, --disassemble        Display assembler contents of executable sections
  -D, --disassemble-all    Display assembler contents of all sections
  -S, --source             Intermix source code with disassembly
  -s, --full-contents      Display the full contents of all sections requested
  -g, --debugging          Display debug information in object file
  -e, --debugging-tags     Display debug information using ctags style
  -G, --stabs              Display (in raw form) any STABS info in the file
  -W[lLiaprmfFsoRtUuTgAckK] or
  --dwarf[=rawline,=decodedline,=info,=abbrev,=pubnames,=aranges,=macro,=frames,
          =frames-interp,=str,=loc,=Ranges,=pubtypes,
          =gdb_index,=trace_info,=trace_abbrev,=trace_aranges,
          =addr,=cu_index,=links,=follow-links]
                           Display DWARF info in the file
  -t, --syms               Display the contents of the symbol table(s)
  -T, --dynamic-syms       Display the contents of the dynamic symbol table
  -r, --reloc              Display the relocation entries in the file
  -R, --dynamic-reloc      Display the dynamic relocation entries in the file
  @<file>                  Read options from <file>
  -v, --version            Display this program's version number
  -i, --info               List object formats and architectures supported
  -H, --help               Display this information
cs304@cs304-devel:~/Desktop/cs1217/A1$ objdump -f main.o

main.o:     file format elf64-x86-64
architecture: i386:x86-64, flags 0x00000011:
HAS_RELOC, HAS_SYMS
start address 0x0000000000000000
```

(d) **file** - file command performs various tests on the file to determine the file type and displays it.