ASSIGNMENT 3 - MAKE

**Q 1)**

Given 4 files main.c , file1.c , file2.c and file3.c with the following dependency:

main.o: file1.c file2.c file3.c

Write a makefile to generate an executable for the program.

**Note:** Create your own c files with simple functionalities.

**Q 2)**

Write a makefile for the set of files and dependencies given in Question 1 that generates the optimised executable code for any architecture.

**Q 3)**

Given a directory comprising some C and C++ files, write a makefile to compile them and store the executables in the two different directories (say “C\_EXE’” and “CPLUS\_EXE’”) depending upon their extensions.

EXAMPLE:

Let there be the following C files in your home directory: 1.c, 2.c, 3.c

and

Let there be the following C++ files in your home directory: 4.cpp, 5.cpp, 6.cpp.

Also let there be “C\_EXE’” and “CPLUS\_EXE’” directories in your home directory.

After executing the make file, executables of C codes should be stored in “C\_EXE” directory and C++ codes in “C++\_EXE” directory.

**Note:** Create the directory structure and C and C++ files. Also, output at the end, the filenames that compiled successfully/unsuccessfully.

**Q 4)**

Given a C/C++ program that takes command line arguments, write a makefile that accepts the arguments (via command line in make) and generates the executable.

**Note:** Write your own C/C++ code that takes command line arguments and does some manipulation on the arguments.

Example:- Consider the following code demo.c

#include<stdio.h>

int main(int agrc, char \*argv[])

{

int X,Y;

X=(int)argv[1];

Y=(int)argv[2];

printf(“%d”,X+Y);

return 0;

}

The makefile should take command line arguments (say, $]make run X=2 Y=3 ) and the executable generated should pass the values (of X and Y) to the C code via command line arguments.

**Q5)**

Given the directory structure as follows:-

Main

Main\_prog.c

Add

Two\_numbers.c

Three\_numbers.c

Sub

Two\_numbers.c

Three\_numbers.c

Mul

Two\_numbers.c

Three\_numbers.c

Div

Two\_numbers.c

Three\_numbers.c

In the directory structure given above, as each Directory comprises different c codes (non dependent) they maybe compiled separately. So they may have their own Makefiles which may be used by the top level makefile to create the executables.

Using the above concept, write the makefiles to compile and generate an executable.

**Note:** Create the above directory structure and the codes, whose functionalities are your own.

**Q 6)**

Given a dependency.txt file which contains the list of files upon which your main program “main\_pr.c” is dependent

FILENAME PATHNAME

file1.c jatinga.iitg.ernet.in/~argha

file2.c jatinga.iitg.ernet.in/~argha

file3.c jatinga.iitg.ernet.in/~argha

**Note:** Create your dependency.txt file as shown above. Obviously the path in jatinga should be directed to your own directory.

Write a make that checks which of files are available on your machine and if not, download them from PATHNAME and hence generate the executable.