**C – PROJECT GROUP 10**

8 – BIT GRAYSCALE PNG TO 8 – BIT GRAYSCALE BMP

TEAM MEMBERS:

1. Ishaan Jalan – IMT2020561

2. Heet Vasani – IMT2020088

3. Siddharth Yedlapati - IMT2020013

4. Dhanvi Medha – IMT2020529

5. Amar Pratap Singh – IMT2020131

6. Pushpang Sanjaykumar Patel - IMT2020104

CONTRIBUTION:

**1. Ishaan Jalan – IMT2020561**

Reading all the chunks of the PNG file . Decompressing the pixel data using zlib.

Wrote the files PNG\_struct.h and PNG\_read.c . (Along with Amar Pratap Singh)

**2. Amar Pratap Singh – IMT2020131**

Reading all the chunks of the PNG file. Doing makefile.

**3. Dhanvi Medha – IMT2020529**

Writing the BMP components into the final image. Wrote the files BMP\_struct.h and BMP\_write.c. (Along with Heet Vasani)

**4. Heet Vasani – IMT2020088**

Writing the BMP components into the final image. Doing makefile.

**5. Siddharth Yedlapati - IMT2020013**

Figured out defilturing of scanlines of the pixel array. Wrote the file filter.c.

**6. Pushpang Sanjaykumar Patel - IMT2020104**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

BUILDING PROCESS

All the files should be in one directory which is required to complete the program.

All the required files are :-

BMP\_struct.h

function.h

PNG\_struct.h

BMP\_write.c

PNG\_read.c

filter.c

main.c

zpipe.c

makefile

libz.a

<image>.png

Composition of makefile

The ‘gcc -c file.c’ command compile file.c file but does not make an executable instead make an object (file.o) file due to ‘-c’ after gcc which can be used later in other commands where they are required.

The general format of coding makefile is:

task: dependencies

command

Output is the task which will be performed by default by using ‘make’ command.

It can also be performed by using ‘make output’ command.

Output depends upon ‘main.o’, ‘PNG\_read.o’, ‘filter.o’, ‘BMP\_write.o’ and ‘zpipe.o’

output: main.o PNG\_read.o filter.o BMP\_write.o zpipe.o

gcc -o my\_project main.o PNG\_read.o filter.o BMP\_write.o libz.a zpipe.o -lz

This gcc command will work only if all the files.o are created successfully. So,

main.o: main.c

gcc -c main.c

PNG\_read.o: PNG\_read.c

gcc -c PNG\_read.c

filter.o: filter.c

gcc -c filter.c

BMP\_write.o: BMP\_write.c

gcc -c BMP\_write.c

zpipe.o : zpipe.c

gcc -I . -c zpipe.c -lz

these are the respective codes to make respective object files.

Since, zpipe.c is linked with zlib.h hence to work, in the command ‘-lz’ extension is added to include zlib header file.

Now, control goes back to our first task ‘output’, where the following command is executed:

gcc -o my\_project main.o PNG\_read.o filter.o BMP\_write.o libz.a zpipe.o -lz

‘-o my\_project’ makes an executable of name “my\_project” (once the above command is executed successfully) which when executed runs the program and a desired <filename>.bmp is created.

One task is ‘clean’:

clean:

rm main.o PNG\_read.o filter.o BMP\_write.o zpipe.o my\_project temp.bin

When this task is performed using ‘make clean’ command, all the executable along with all the .o files is removed. From the outputs, only the finalimage (bmp format) is retained.