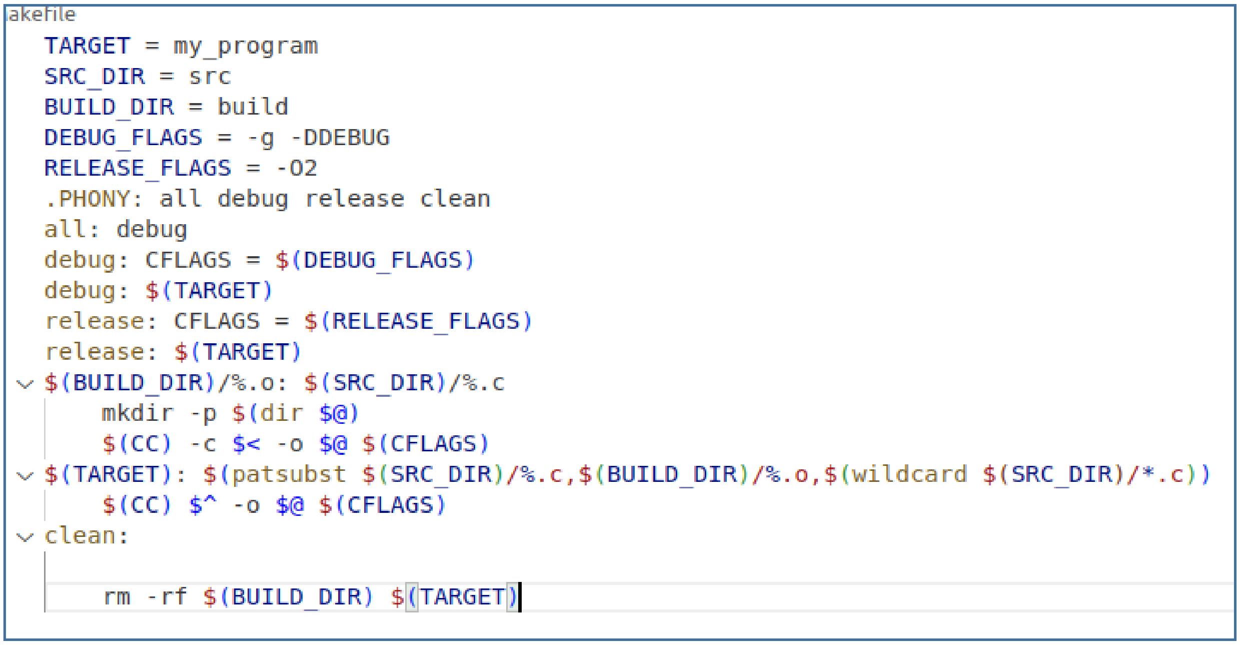
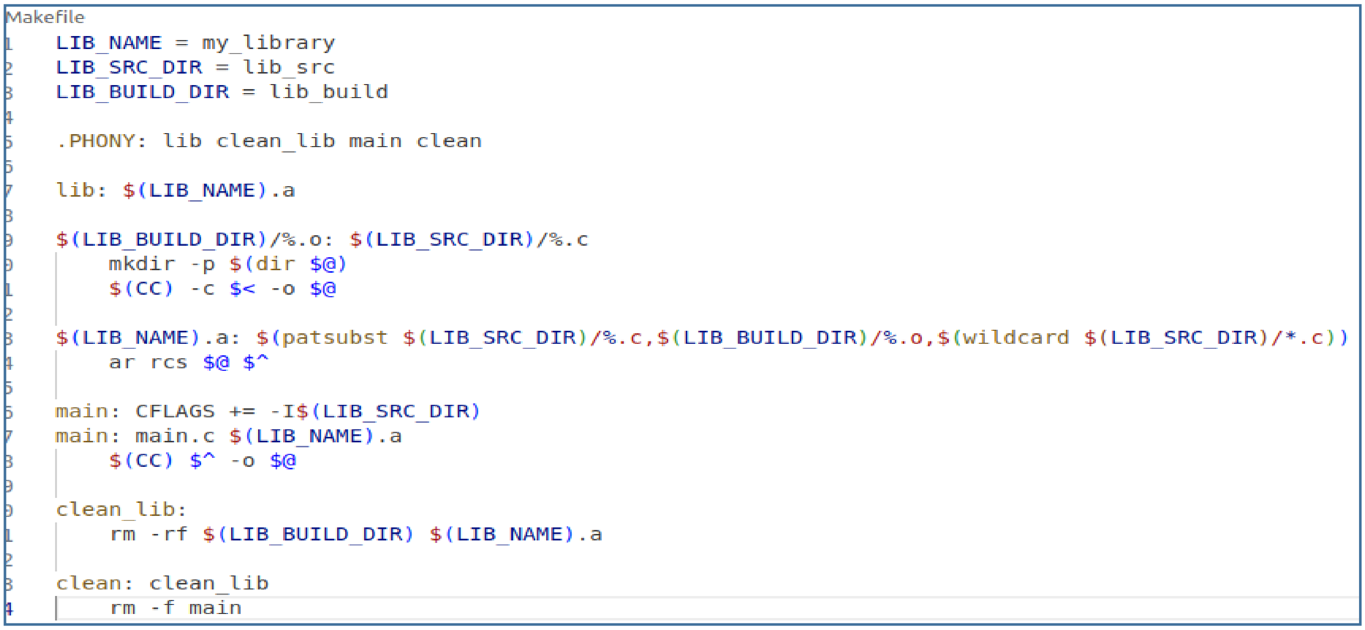
LAB 12

Q1:



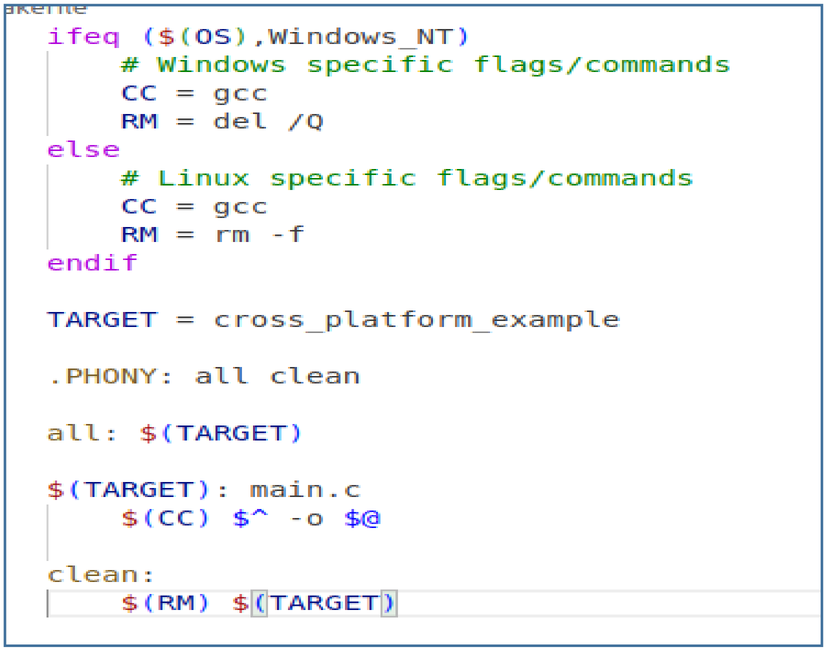
```
TARGET = my_program
SRC_DIR = src
BUILD_DIR = build
DEBUG_FLAGS = -g -DDEBUG
RELEASE_FLAGS = -O2
.PHONY: all debug release clean
all: debug
debug: CFLAGS = $(DEBUG_FLAGS)
debug: $(TARGET)
release: CFLAGS = $(RELEASE_FLAGS)
release: $(TARGET)
$(BUILD_DIR)/%.o: $(SRC_DIR)/%.c
    mkdir -p $(dir $@)
    $(CC) -c $< -o $@ $(CFLAGS)
$(TARGET): $(patsubst $(SRC_DIR)/%.c,$(BUILD_DIR)/%.o,$(wildcard $(SRC_DIR)/*.c))
    $(CC) $^ -o $@ $(CFLAGS)
clean:

    rm -rf $(BUILD_DIR) $(TARGET)
```

Q2:



```
LIB_NAME = my_library
LIB_SRC_DIR = lib_src
LIB_BUILD_DIR = lib_build

.PHONY: lib clean_lib main clean

lib: $(LIB_NAME).a

$(LIB_BUILD_DIR)/%.o: $(LIB_SRC_DIR)/%.c
    mkdir -p $(dir $@)
    $(CC) -c $< -o $@

$(LIB_NAME).a: $(patsubst $(LIB_SRC_DIR)/%.c,$(LIB_BUILD_DIR)/%.o,$(wildcard $(LIB_SRC_DIR)/*.c))
    ar rcs $@ $^

main: CFLAGS += -I$(LIB_SRC_DIR)
main: main.c $(LIB_NAME).a
    $(CC) $^ -o $@

clean_lib:
    rm -rf $(LIB_BUILD_DIR) $(LIB_NAME).a

clean: clean_lib
    rm -f main
```

Q3:



```
ifeq ($(OS),Windows_NT)
    # Windows specific flags/commands
    CC = gcc
    RM = del /Q
else
    # Linux specific flags/commands
    CC = gcc
    RM = rm -f
endif

TARGET = cross_platform_example

.PHONY: all clean

all: $(TARGET)

$(TARGET): main.c
    $(CC) $^ -o $@

clean:
    $(RM) $(TARGET)
```