```
PROJECT_NAME = Sortingalgo

SRC = src/BubbleSort.c\
src/InsertionSort.c\
src/QuickSort.c\
src/MergeSort.c

GCNO = BubbleSort\
InsertionSort\
QuickSort\
MergeSort

BUILD = build

#INC=inc
UNI=unity/unity.c

INC   = -Iunity\
-Itest\

ifdef OS
   RM = del /q
   FixPath = $(subst /,\,$1)
   EXEC = exe
else
   ifeq ($(shell uname), Linux)
      RM = rm -rf
      FixPath = $1
        EXEC = out
   endif
endif

TEST_SRC = test/test_sorting

PROJECT_OUTPUT = $(BUILD)/$(PROJECT_NAME).out

TEST_OUTPUT = $(BUILD)/Test_$(PROJECT_NAME).out

.PHONY: run clean test doc


$(PROJECT_NAME).exe : $(SRC)
      gcc -I inc $(INC) $(PROJECT_NAME).c $(SRC) main.c -o
$(PROJECT_NAME).exe

run: $(PROJECT_NAME).exe
      $(PROJECT_NAME).exe

test:$(TEST)
      gcc -Iinc $(INC) $(UNI) $(SRC) $(TEST_SRC).c -o test\Main.exe
      ./test\Main.exe

coverage:
      gcc -fprofile-arcs -ftest-coverage -Iinc $(INC) $(UNI) $(SRC)
$(TEST_SRC).c $(PROJECT_NAME).c -o all.out
      ./all.out
      gcov $(GCNO).gcno
```

```
#cat *.c.gcov

clean:
	del *gcda *gcov *gcno

$(BUILD):
	mkdir build

all: $(SRC) $(BUILD)
	gcc -I inc $(UNI) $(SRC) $(INC) $(TEST_SRC).c -o $(PROJECT_OUTPUT)

run1:
	./$(PROJECT_OUTPUT)
```