Assignment 3 Result Report
通訊二 110503531 劉泰君

1.Compile Result

- Arithmetic

```
cornertw@DESKTOP-HD1IFP1:/mnt/c/Users/Max Liu/NCU_Documents/arcd$ make
[ 25%] Built target arcd
[ 50%] Built target adaptive_model
Scanning dependencies of target arcd_stream
[ 62%] Building C object examples/CMakeFiles/arcd_stream.dir/arcd_stream.c.o
[ 75%] Linking C executable arcd_stream
[ 75%] Built target arcd_stream
Scanning dependencies of target codec_tests
[ 87%] Building CXX object tests/CMakeFiles/codec_tests.dir/codec_tests.cpp.o
[100%] Linking CXX executable codec_tests
[100%] Built target codec_tests
```

- Huffman

```
cornertw@DESKTOP-HD1IFP1:/mnt/c/Users/Max Liu/NCU_Documents/huffman-main$ make
cc -g -Wall -Werror -Wextra -O0 -std=c11 -D_POSIX_C_SOURCE=2   -c -o huffcode.o huffcode.c
cc  -o huffcode huffcode.o libhuffman.a
```

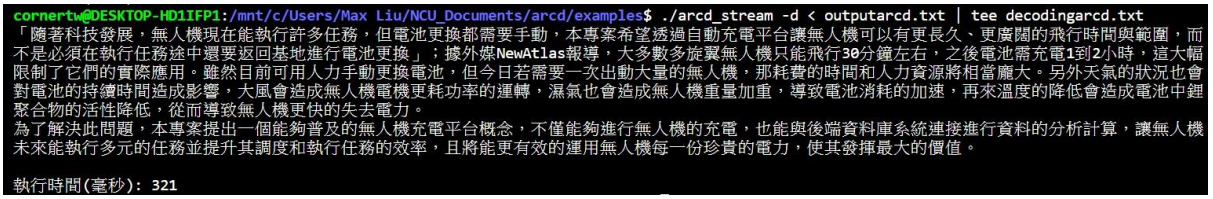
2.Execute Result

Arithmetic

- Encoding
- input file is “runarcd.txt” ,
- encoding file is “outputarcd.txt”

```
cornertw@DESKTOP-HD1IFP1:/mnt/c/Users/Max Liu/NCU_Documents/arcd/examples$ ./arcd_stream -e < runarcd.txt | tee outputarcd.txt
:d>Bs|yY▯▯wt*ǰWKxl?
                    F▯{6Jno
                          d7DH4Ɣ
;zI▯^'`bcen%=LjPV:2e@ns'ɾo(*         YRum&[j]g#u|]0dYXw+?#Q       Bo
                                                                  Ɣ/_Kй⊥_;r/^ô2}T3QHl~{fË▯:J;a!xl4MAMRQf~
                                                                                                        1
(C:C=d#Ƒn!Z-T:*ÿbA$n+U|X-7Sc
ZE`UX +6saw1Bf*
HgzY@~0wt5t:AP_ IxOƃxBZGEWbNȝ8}Io␣Uhhr    3}ʜuOz7%nŭQnqJ_ŧvU*ĥTf;ybY$JQCm̃l*&0C8O彡 mo<n
▯ h"壛ɬ7E~yIb:h>!U       <7Cn7P?ZvyhBD[{+-q30eO9x
                                        $ JF{
                                             RьjYƺ^:     ~7*Yqka▯Ysm:OJ]ią̈ÿ,~kJ`f[oz@48`!mi;
                                                                                                @A;'D uz(3M.PwHgV^Ir
!RloQikF      ▯ɣmW]bVO-61"/
                          /:T$V17
M\"P\>3w˝pN:
執行時間(毫秒): 238
```

- Decoding
- input file is “outputarcd.txt” ,
- decoding file is “decodingarcd.txt”



Huffman

- Encoding
- input file is “runhuff.txt” , encoding file is “outputhuff.txt”

- Decoding
- input file is “outputhuff.txt” ,decoding file is “runhuff.txt”

3.Analysis

Because one of my classmates and I coded with longer data content and the other with shorter data, I found that Huffman coding is faster when processing large data, while arithmetic coding is faster when processing a small amount of data.