Getting to Know the 185 Environment

LAB 1

SECTION 3

James Gaul

Submission Date:

1/28/24

Lab Date:

1/26/24

Problem

The purpose of this lab is to gain familiarity with Cygwin, learning how to navigate between locations, list the files present, and compile C codes. Following this, the lab involves a demonstration of how to execute, modify, and implement C code. This includes various demonstrations of various input/output options, including how to read/write from a text file and how to link separate programs into a single process.

Analysis

As I have prior experience in a command line Unix environment, I experienced very little difficulty in running the programs. The programs compiled and ran smoothly, demonstrating a number of useful capabilities for future labs and other projects.

Design

Not applicable to this lab, as the code we were to use was already provided.

Testing

All three programs ran without difficulty.

Comments

Nothing more to add.

Source Code

lab01-1.c

```c
/*------------------------------------------------------------------------------
-           SE 185: Lab 01 - Getting to Know the 185 Environment             -
-   Name: James Gaul
-   Section: 3                                                               -
-   NetID: 947125207
-   Date: 1/26/24
------------------------------------------------------------------------------*/

/*
    Here is a block comment.
    These lines don't run when you compile or run the code.
    I.e., they're the machine doesn't see them, only you.
*/

/*------------------------------------------------------------------------------
-                               Includes                                       -
------------------------------------------------------------------------------*/
#include <stdio.h>

/*------------------------------------------------------------------------------
-                               Implementation                                 -
------------------------------------------------------------------------------*/
int main(int argc, char *argv[])
{
    // This is a C comment, this line doesn't run in the program

    printf("Test sentence."); // Modify this line
    printf("\n"); // This prints a newline character

    return 0; // This is a return statement
}
```

lab01-input.c

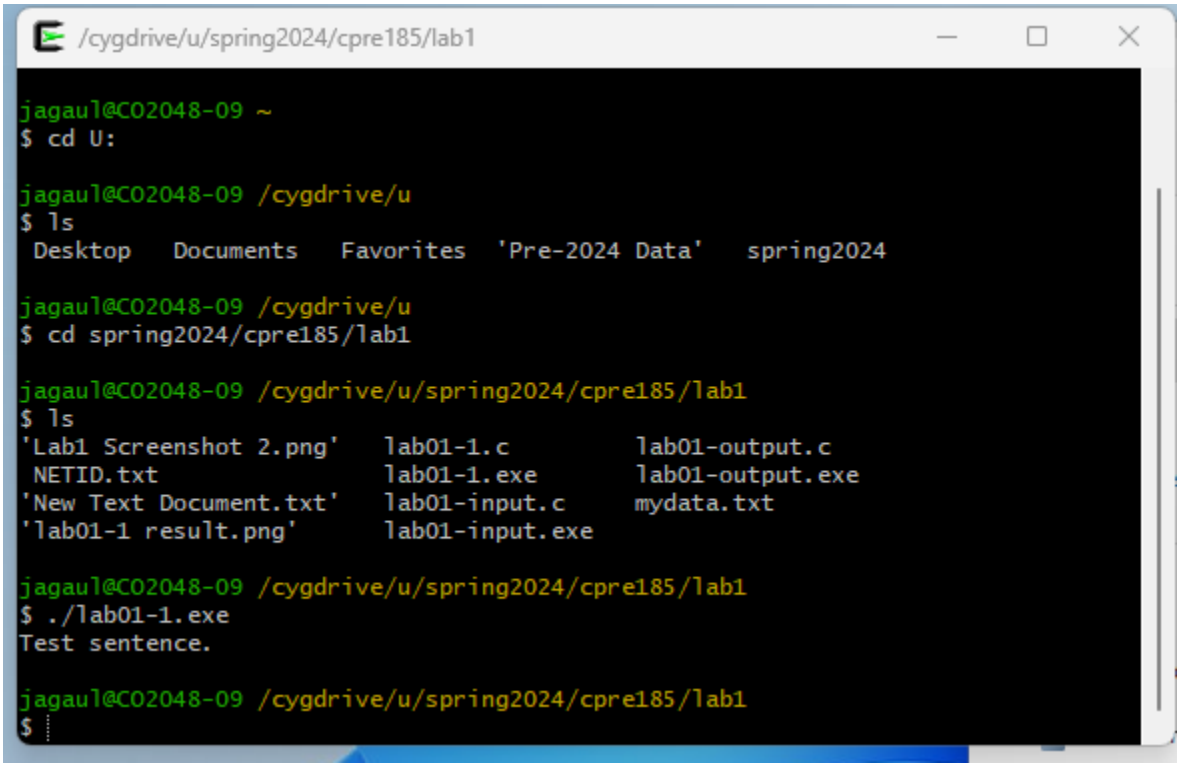
```c
/*------------------------------------------------------------------------------
-           SE 185: Lab 01 - Getting to Know the 185 Environment             -
-   Name: James Gaul                                                         -
-   Section: 3                                                           -
-   NetID: 947125207                                                         -
-   Date: 1/26/24                                                        -
------------------------------------------------------------------------------*/

/*------------------------------------------------------------------------------
-                               Includes                                       -
------------------------------------------------------------------------------*/
#include <stdio.h>
#include <stdlib.h>

/*------------------------------------------------------------------------------
-                               Implementation                                 -
------------------------------------------------------------------------------*/
int main(int argc, char *argv[])
{
    /* DO NOT EDIT THIS FILE */
    char net_id[1000];

    int number = 0;

    printf("\nValue before input: %d\n", number);

    printf("\nType a number: ");
    scanf("%d", &number);

    printf("Type your NetID: ");
    scanf("%s", net_id);

    printf("\nYour input was %d ", number);
    printf("and your ISU email is %s@iastate.edu", net_id);

    return 0;
}
```

lab01-output.c

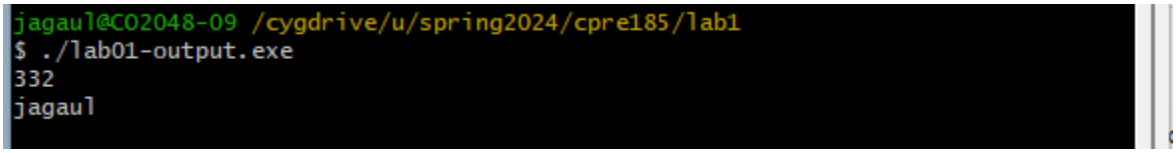
```c
/*-----------------------------------------------------------------------------
-           SE 185: Lab 01 - Getting to Know the 185 Environment             -
-   Name: James Gaul
-   Section: 3                                                      -
-   NetID: 947125207
-   Date: 1/26/24                                                       -
-----------------------------------------------------------------------------*/

/*-----------------------------------------------------------------------------
-                               Includes                                     -
-----------------------------------------------------------------------------*/
#include <stdio.h>

/*-----------------------------------------------------------------------------
-                               Implementation                               -
-----------------------------------------------------------------------------*/
int main(int argc, char *argv[])
{
    int number = 332; // Change the zero to a different number.

    printf("%d", number);
    printf("\n");

    printf("jagaul"); // Change this to your NetID.
    printf("\n");

    return 0;
}
```

Screenshots

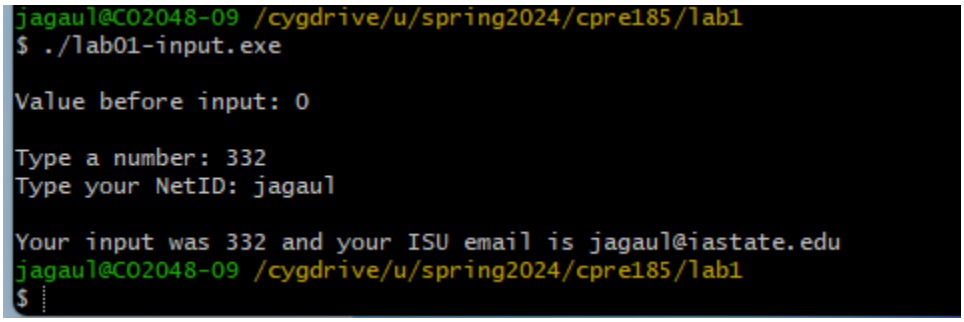
Screenshot 1:



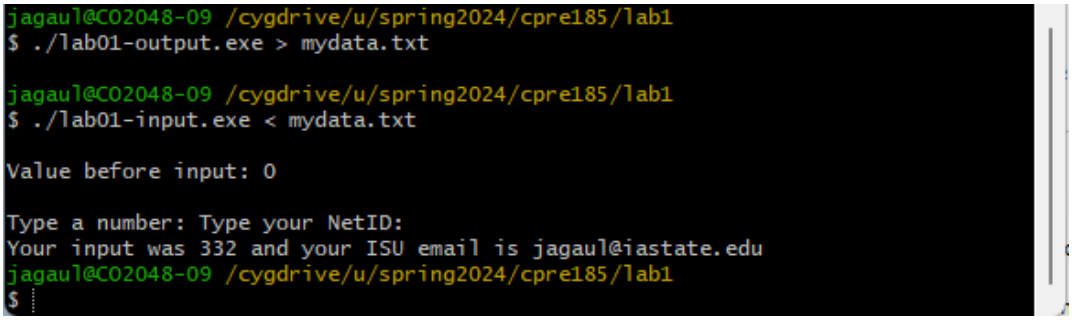
Screenshot 2:



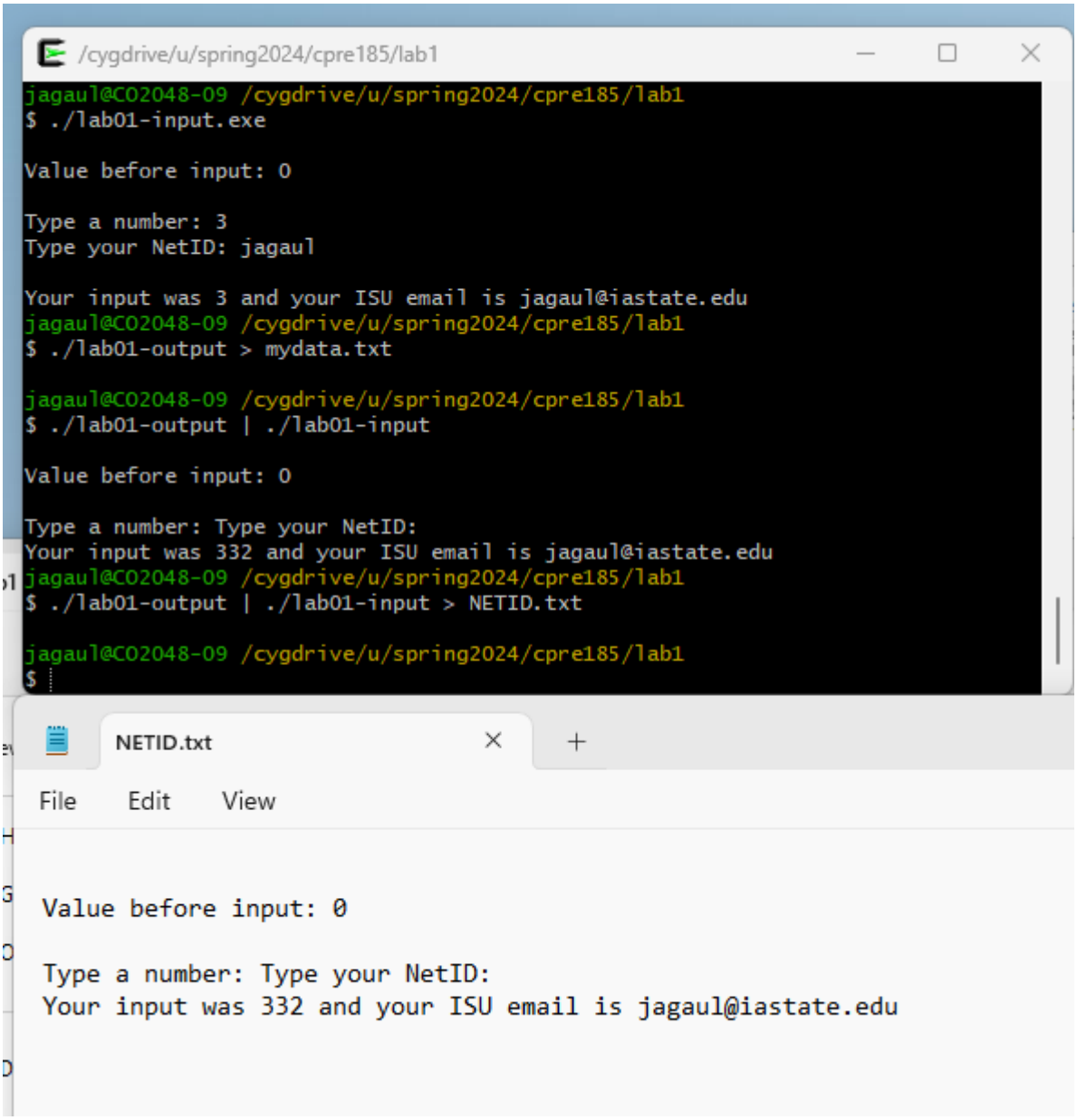
Screenshot 3



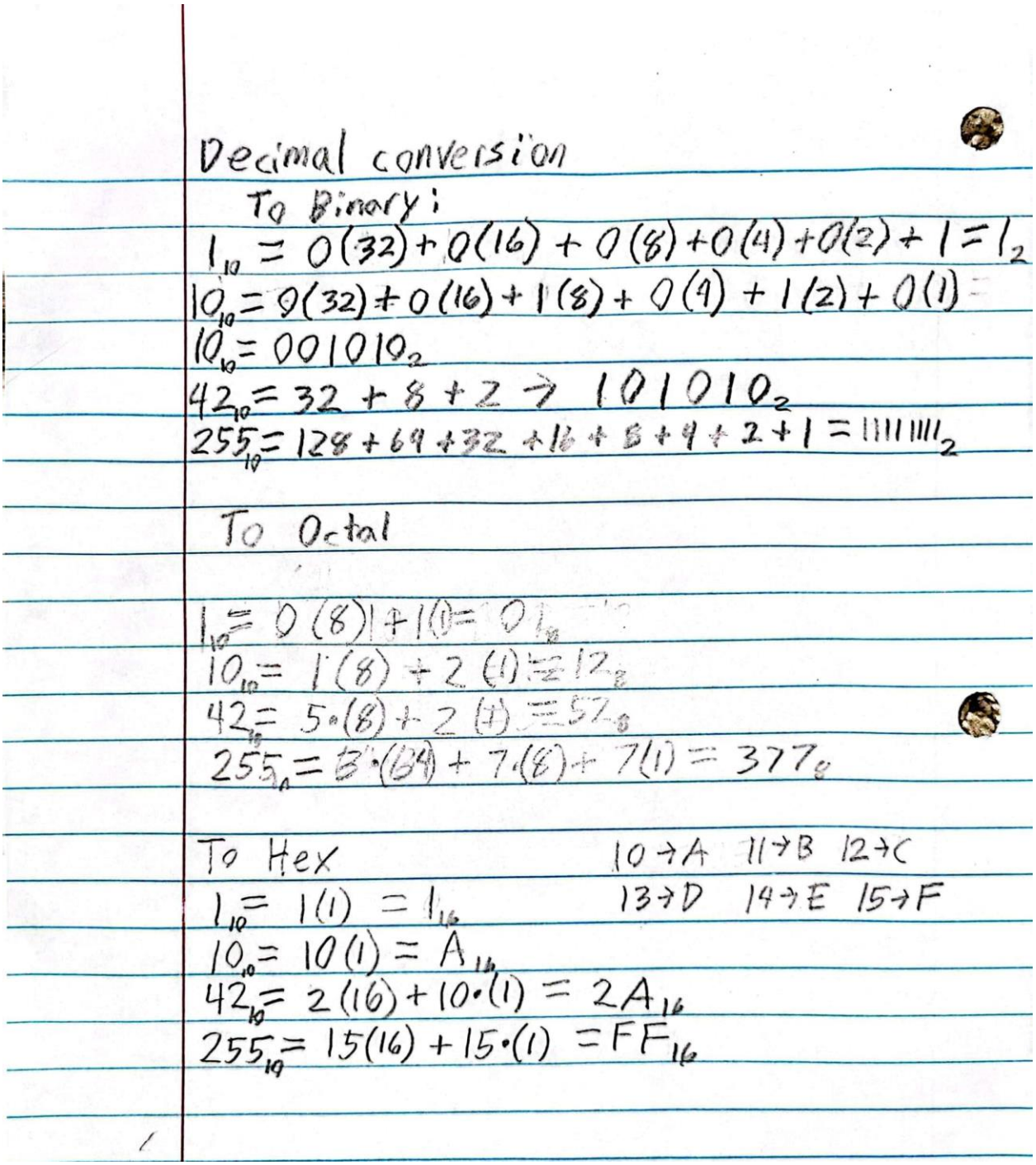
Screenshot 4:



Screenshot 5:



Conversion Problems:



Decimal conversion

To Binary:

$1_{10} = 0(32) + 0(16) + 0(8) + 0(4) + 0(2) + 1 = 1_2$

$10_{10} = 0(32) + 0(16) + 1(8) + 0(4) + 1(2) + 0(1) =$

$10_{10} = 001010_2$

$42_{10} = 32 + 8 + 2 \rightarrow 101010_2$

$255_{10} = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 11111111_2$

To Octal

$1_{10} = 0(8) + 1(1) = 01_8$

$10_{10} = 1(8) + 2(1) = 12_8$

$42_{10} = 5 \cdot (8) + 2(1) = 52_8$

$255_{10} = 3 \cdot (64) + 7 \cdot (8) + 7(1) = 377_8$

To Hex

$10 \rightarrow A \quad 11 \rightarrow B \quad 12 \rightarrow C$

$13 \rightarrow D \quad 14 \rightarrow E \quad 15 \rightarrow F$

$1_{10} = 1(1) = 1_{16}$

$10_{10} = 10(1) = A_{16}$

$42_{10} = 2(16) + 10 \cdot (1) = 2A_{16}$

$255_{10} = 15(16) + 15 \cdot (1) = FF_{16}$

Binary Conversion:

To Decimal:

a. $10010011_2 = 1(128)+0(64)+0(32)+1(16)+0(8)+0(4)+1(2)+1($
$10010011_2 = 147_{10}$

b. $111111_2 = 1(32)+1(16)+1(8)+1(4)+1(2)+1(1) = 63_{10}$

To Hexadecimal

a. $147_{10} = 9(16)+3(1) = 93_{16}$

b. $63_{10} = 3(16)+15(1) = 3F$

T. To Octal

a. $147_{10} = 2(64)+2(8)+3(1) = 223_8$

b. $63_{10} = 7(8)+7(1) = 77_8$

Hexadecimal conversion

A→10 B→11 C→12
D→13 E→14 F→15

To Decimal:

a. $F_{16} = 15 \cdot (1) = 15_{10}$
b. $DF_{16} = 13(16) + 15(1) = 223_{10}$
c. $81_{16} = 8(16) + 1(1) = 129_{10}$

To Binary (going from previously found decimal)

a. $15_{10} = 0 \cdot (16) + 1(8) + 1(4) + 1(2) + 1(1) = 01111_2$
b. $223_{10} = 1(128) + 1(64) + 0(32) + 1(16) + 1(8) + 1(4) + 1(2) + 1(1)$
$223_{10} = 11011111_2$
c. $129_{10} = 1(128) + 0(64) + 0(32) + 0(16) + 0(8) + 0(4) + 0(2) + 1(1)$
$129_{10} = 10000001$

To Octal:

a. $15_{10} = 1(8) + 7(1) = 17_8$
b. $223_{10} = 3(64) + 3(8) + 7(1) = 337_8$
c. $129_{10} = 2(64) + 0(8) + 1(1) = 201_8$

Octal Conversion

To Decimal:

$22_8 = 2(8) + 2(1) = 18_{10}$

To Binary:

$18_{10} = 1(16) + 0(8) + 0(4) + 1(2) + 0(1) = 10010_2$

To Hexadecimal:

$18_{10} = 1(16) + 2(1) = 12_{16}$