프로그래밍기초 기말시험(40점)

■ 주의 사항
- ✓ 소스 파일 상단에 학번, 이름을 주석으로 추가하세요.
  - 주석이 없으면, 파일당 -1점
- ✓ 컴파일 에러가 발생하면 0점 처리함
- ✓ 실행 결과를 하드 코딩하면 0점 처리함
- ✓ 문제에서 제시한 조건을 따르지 않는 경우, 해당 문제의 점수는 없음
- ✓ 지각 제출시 본인이 받은 점수에서 1분당 10%씩 감점함

1. 두 파일의 내용 비교 및 라인 번호 출력 프로그램 (10점)

- 제출 파일: final01.c

2개의 텍스트 파일을 라인 단위로 비교하고 서로 다른 부분에 대해서는 라인 번호 및 파일의 내용을 출력하는 프로그램을 작성하시오. 파일의 길이가 서로 다른 경우에는 반드시 파일의 끝까지 검사해서 화면에 출력해야 됨.

사용 파일: sample1.txt, sample2.txt, sample3.txt

■ 동일한 파일 비교 기능 (2점)
- ✓ 두 개의 파일을 읽기 모드로 열고 한 라인씩 파일 내용을 비교함
- ✓ 일치하는 부분은 화면에 출력하지 않고 파일의 끝까지 비교해서 동일하면 “The two files are identical.” 만 출력 후 프로그램 종료함
- ✓ 동일한 파일 이름을 입력한 경우에도 파일의 끝까지 비교하는 기능을 반드시 구현해야 됨: 파일 비교 기능이 없으면 0점 처리함

■ 서로 길이 및 내용이 다른 파일 비교 기능 (8점, 각 4점)
- ✓ 서로 다른 부분을 발견하면 각 파일의 실제 파일의 라인 번호 및 파일 내용을 화면에 출력함 (아래 예시 내용 참조)
  - << 13: Objective-C, Objective-C++, Fortran, Ada, Go, D, Modula-2, Rust and COBOL among
  - >> 13: Objective-C++, Fortran, Go, D, Modula-2, Rust and COBOL among others.[6]
- ✓ 두 파일의 길이가 다른 경우에는, 긴 파일의 라인 번호 및 파일 내용을 모두 화면에 출력

■ 실행 결과 1: 동일한 파일 이름을 2개 입력한 경우

```
Type the first file name: sample1.txt
Type the second file name: sample1.txt
The two files are identical.
```

■ 서로 다른 파일 비교 #2 (sample2.txt, sample1.txt 등 다양한 순서로 파일 이름 입력 후 테스트 진행)

■ 실행 결과 2: 서로 다른 파일 비교 #1 (sample2.txt, sample1.txt)

```
Type the first file name: sample2.txt
Type the second file name: sample1.txt
<< 13: Objective-C++, Fortran, Go, D, Modula-2, Rust and COBOL among others.[6]
>> 13: Objective-C, Objective-C++, Fortran, Ada, Go, D, Modula-2, Rust and COBOL among
<< 14: The OpenMP and OpenACC specifications are also supported in the C and C++
>> 14: others.[6] The OpenMP and OpenACC specifications are also supported in the C and C++
>> 24:      --> sample1.txt 파일에만 존재하는 내용 출력
>> 25: GCC has been ported to more platforms and instruction set architectures than any
>> 26: other compiler, and is widely deployed as a tool in the development of both free and
>> 27: proprietary software. GCC is also available for many embedded systems, including
>> 28: ARM-based and Power ISA-based chips.
```

■ 실행 결과 2: 서로 다른 파일 비교 #2 (sample1.txt, sample3.txt)

```
Type the first file name: sample1.txt
Type the second file name: sample3.txt
<<  1: The GNU Compiler Collection (GCC) (formerly GNU C Compiler) is a collection of
>>  1: ANSI C and ISO C
<<  2: compilers from the GNU Project that support various programming languages, hardware
>>  2: During the late 1970s and 1980s,
<<  3: architectures, and operating systems. The Free Software Foundation (FSF) distributes
>>  3: versions of C were implemented for a wide variety of mainframe computers,
<<  4: GCC as free software under the GNU General Public License (GNU GPL). GCC is a key
>>  4: minicomputers, and microcomputers, including the IBM PC,
<<  5: component of the GNU toolchain which is used for most projects related to GNU and
>>  5: as its popularity began to increase significantly.
<<  6: the Linux kernel. With roughly 15 million lines of code in 2019, GCC is one of the
>>  6: In 1983, the American National Standards Institute (ANSI) formed a committee,
<<  7: largest free programs in existence.[4] It has played an important role in the growth
>>  7: X3J11, to establish a standard specification of C.
<<  8: of free software, as both a tool and an example.
>>  8: X3J11 based the C standard on the Unix implementation;
<<  9:
>>  9: however, the non-portable portion of the Unix C library was handed off to the IEEE
<< 10: When it was first released in 1987 by Richard Stallman, GCC 1.0 was named the GNU C
>> 10: working group 1003
<< 11: Compiler since it only handled the C programming language.[1] It was extended to
>> 11: to become the basis for the 1988 POSIX standard.
<< 12: compile C++ in December of that year. Front ends were later developed for
>> 12: In 1989, the C standard was ratified as ANSI X3.159-1989 "Programming Language C".
<< 13: Objective-C, Objective-C++, Fortran, Ada, Go, D, Modula-2, Rust and COBOL among
>> 13: This version of the language is often referred to as ANSI C, Standard C, or
<< 14: others.[6] The OpenMP and OpenACC specifications are also supported in the C and C++
>> 14: sometimes C89.
<< 15: compilers.[7][8]
>> 15: In 1990, the ANSI C standard (with formatting changes) was adopted by the
<< 16:
>> 16: International Organization
```

```
<< 17: As well as being the official compiler of the GNU operating system, GCC has been
>> 17: for Standardization (ISO) as ISO/IEC 9899:1990, which is sometimes called C90.
<< 18: adopted as the standard compiler by many other modern Unix-like computer operating
>> 18: Therefore, the terms "C89" and "C90" refer to the same programming language.
<< 19: systems, including most Linux distributions. Most BSD family operating systems also
>> 19:
<< 20: switched to GCC shortly after its release, although since then, FreeBSD and Apple
>> 20: GCC was first released March 22, 1987, available by FTP from MIT.[18] Stallman was
<< 21: macOS have moved to the Clang compiler,[9] largely due to licensing
>> 21: listed as the author but cited others for their contributions, including Tower for
<< 22: reasons.[10][11][12] GCC can also compile code for Windows, Android, iOS, Solaris,
>> 22: "parts of the parser, RTL generator, RTL definitions, and of the Vax machine
<< 23: HP-UX, AIX, and MS-DOS compatible operating systems.[13]
>> 23: description", Jack Davidson and Christopher W. Fraser for the idea of using RTL as
<< 24:
>> 24: an intermediate language, and Paul Rubin for writing most of the preprocessor.[19]
<< 25: GCC has been ported to more platforms and instruction set architectures than any
>> 25: Described as the "first free software hit" by Peter H. Salus, the GNU compiler
<< 26: other compiler, and is widely deployed as a tool in the development of both free and
>> 26: arrived just at the time when Sun Microsystems was unbundling its development tools
<< 27: proprietary software. GCC is also available for many embedded systems, including
>> 27: from its operating system, selling them separately at a higher combined price than
<< 28: ARM-based and Power ISA-based chips.
>> 28: the previous bundle, which led many of Sun's users to buy or download GCC instead of
>> 29: the vendor's tools.[20] While Stallman considered GNU Emacs as his main project, by
>> 30: 1990 GCC supported thirteen computer architectures, was outperforming several vendor
>> 31: compilers, and was used commercially by several companies.[21]
```

2. 파일 입출력을 이용한 도서 관리 프로그램 (30점)

- 제출 파일: final02.c

주어진 파일(books.txt)파일을 읽고 구조체 배열에 내용을 저장한 다음, 책 정보 추가, 검색, 업데이트 및 최종 결과를 저장하는 프로그램을 구현하세요.

■ 구조체 정보

```c
#define MAX_BOOK 10

typedef struct {
    char title[50];
    char author[30];
    int price;
} book;

// 전역 변수
int book_count = 0;  // 전체 구조체 배열에 저장된 책 개수
int main()
{
    book booklib[MAX_BOOK] = {0};
    . . .
    return 0;
}
```

- ✓ book booklib[MAX_BOOK] 구조체 배열은 반드시 main() 함수에서 선언한 다음 각 함수로 포인터 전달 방식으로 사용해야 됨 (전역 변수로 사용시 20점 감점 함)

■ **books.txt파일 읽고 구조체 배열에 저장 (2점)**

- ✓ 파일을 읽기 모드로 읽어서 구조체 배열 booklib[]에 저장
  - 파일의 각 항목들은 콤마(,)로 구분되어 있음
  - 아래의 샘플 코드 사용 (strtok 사용도 가능)
  - 파일을 읽고 저장된 구조체의 개수 화면에 출력
  - 출력 예: Read books.txt (8 books) successfully.

```
char line[100] = {0};
. . .
// 파일의 끝까지 한 라인씩 읽고 구조체 배열에 저장
fgets(line, sizeof(line), fin);
sscanf(line, "%[^,],%[^,],%d",
         booklib[index].title,
         booklib[index].author,
         &booklib[index].price);
```

■ 메뉴 구현 (28점)

- ✓ 1. Print All Books 함수 구현 **(4점)**
  - 구조체 배열에 저장된 각 항목들을 화면에 출력
  - 실행 결과 참조
- ✓ **2. Add New Book 함수 구현 (4점)**
  - 최대 10개까지 구조체 배열에 도서 정보 추가 가능
  - title, author에는 공백이 저장 가능해야 되며 도서 정보 추가 후 현재 데이터 개수 출력 (실행 결과 참조)
  - 10개의 데이터가 저장되어 있으면 더 이상 저장 안되면, “No empty space.(10 Books)” 화면 출력
  - 도서 저장 후 저장된 데이터는 메뉴 1번을 통해서 확인되어야 함
- ✓ **3. Search Book 구현 함수 구현 (8점, 부분 점수 없음)**
  - 검색 조건: title에 입력된 키워드를 포함하며 가격 범위에 포함하는 모든 책을 화면에 출력
  - 전체 검색된 개수 화면 출력
  - 입력된 키워드를 포함하는 title이 없거나, 가격 범위에 해당하는 책이

없으면: “No Search Result(검색어, 가격 범위 출력)”

- ✓ **4. Update Book Information 함수 구현 (10점)**
  - – 변경할 index 입력 (1~book_count 범위)
  - – index가 배열의 범위를 넘으면 “Out of index” 출력 후 함수 빠져나감
  - – 정상 범위의 index인 경우, 해당 index의 데이터 출력 (1점)
  - – 변경할 title 항목과 author에 Enter키만 입력한 경우 (4점)
    - title, author 항목은 기존 데이터를 유지
    - price 항목은 반드시 입력
  - – 변경할 title, author, price에 새로운 데이터를 입력한 경우 (4점)
    - 입력된 title, author, price 항목으로 기존 데이터 변경
  - – 변경 후 변경된 데이터 출력 (1점)
- ✓ **5. Save Book Data and Exit Program (2점)**
  - – 현재 변경된 구조체 배열의 전체 내용을 book_out.txt 파일로 저장
    - 파일로 저장된 전체 데이터의 개수 출력
    - VS code에서 book_out.txt 파일 확인

■ 실행 결과

```
Read books.txt (8 books) successfully.  ←---- 프로그램이 시작되면 파일 읽기
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 1
Total Book Count: 8
[1] Title: C Programming, Author: Dennis Ritchie, Price: 18000
[2] Title: Modern C Programming, Author: K.N.King, Price: 35000
[3] Title: Java How to Program, Author:  Paul Deitel, Price: 48000
[4] Title: C++ How to Program, Author:  Dan Quirk, Price: 45000
[5] Title: Understanding Linux Programming, Author: Bruce Molay, Price: 29000
[6] Title: Hands on Machine Learning, Author: John Anderson, Price: 21000
[7] Title: Learning Python, Author: Guido Van Rossum, Price: 27000
[8] Title: Effective Java, Author: Joshua Bloch, Price: 32000


====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
```

```
====================================
Type a number: 2
Type a new book title(Max: 50 chars):  Machine Learning
Type a new book author(Max: 30 chars): Andreas Muller
Type a new book price: 3300
A new book added.(book count: 9)  ←---- 구조체 배열에 저장된 전체 책의 개수 출력
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 2
Type a new book title(Max: 50 chars): Linux Programming
Type a new book author(Max: 30 chars): Neil Matthew
Type a new book price: 39000
A new book added.(book count: 10)
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 2
No empty space.(10 Books)    ←------ 10 개의 데이터가 저장되어 있어 더 이상 저장 안됨
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 1            ←-------- 저장된 전체 데이터 출력
Total Book Count: 10
[1] Title: C Programming, Author: Dennis Ritchie, Price: 18000
[2] Title: Modern C Programming, Author: K.N.King, Price: 35000
[3] Title: Java How to Program, Author:  Paul Deitel, Price: 48000
[4] Title: C++ How to Program, Author:  Dan Quirk, Price: 45000
[5] Title: Understanding Linux Programming, Author: Bruce Molay, Price: 29000
[6] Title: Hands on Machine Learning, Author: John Anderson, Price: 21000
[7] Title: Learning Python, Author: Guido Van Rossum, Price: 27000
[8] Title: Effective Java, Author: Joshua Bloch, Price: 32000
[9] Title: Machine Learning, Author: Andreas Muller, Price: 3300
```

```
[10] Title: Linux Programming, Author: Neil Matthew, Price: 39000
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 3
Type a title keyword for search: Program         ←--title 에 “Program”을 포함
Type a price range(ex: 10000 20000): 15000 35000  ←-- price 범위에 포함하는 도서 검색
[1] Title: C Programming, Author: Dennis Ritchie, Price: 18000
[2] Title: Modern C Programming, Author: K.N.King, Price: 35000
[5] Title: Understanding Linux Programming, Author: Bruce Molay, Price: 29000
Total 3 books found

====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 3
Type a title keyword for search: Java
Type a price range(ex: 10000 20000): 10000 20000
No Search Result(Java, 10000 ~ 20000) ←----- price 범위의 도서가 없는 경우

====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 3
Type a title keyword for search: AI
Type a price range(ex: 10000 20000): 10000 50000
No Search Result(AI, 10000 ~ 50000) ←----- “AI”를 포함하는 title 없음
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 4
Type a index for updating: 11      ←----- 인덱스 범위 밖 입력(1~book_count 범위)
Out of index
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 4
Type a index for updating: 0        ←----- 인덱스 범위 밖 입력(1~book_count 범위)
Out of index
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 4
Type a index for updating: 1
[1] Title: C Programming, Author: Dennis Ritchie, Price: 18000  ←- 해당 데이터 출력
Enter a new title(Enter: skip):           ←---- title 에 Enter 키만 입력
Enter a new author name(Enter: skip):     ←---- name 에 Enter 키만 입력
Enter a new book price: 1800
Updated Successfully                       ←------- price 만 변경됨
[1] Title: C Programming, Author: Dennis Ritchie, Price: 1800 ←-- 변경 내용 출력
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 4
Type a index for updating: 10
[10] Title: Linux Programming, Author: Neil Matthew, Price: 39000
Enter a new title(Enter: skip):  C# Programming
Enter a new author name(Enter: skip): Hong Gil
Enter a new book price: 45000
Updated Successfully
[10] Title: C# Programming, Author: Hong Gil, Price: 45000
```

```
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 1
Total Book Count: 10
[1] Title: C Programming, Author: Dennis Ritchie, Price: 1800
[2] Title: Modern C Programming, Author: K.N.King, Price: 35000
[3] Title: Java How to Program, Author:  Paul Deitel, Price: 48000
[4] Title: C++ How to Program, Author:  Dan Quirk, Price: 45000
[5] Title: Understanding Linux Programming, Author: Bruce Molay, Price: 29000
[6] Title: Hands on Machine Learning, Author: John Anderson, Price: 21000
[7] Title: Learning Python, Author: Guido Van Rossum, Price: 27000
[8] Title: Effective Java, Author: Joshua Bloch, Price: 32000
[9] Title: Machine Learning, Author: Andreas Muller, Price: 3300
[10] Title: C# Programming, Author: Hong Gil, Price: 45000

====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 5
10 Book Data Saved.
Exit Program
```

- ■ book_out.txt 파일 내용

```
C Programming,Dennis Ritchie,1800
Modern C Programming,K.N.King,35000
Java How to Program, Paul Deitel,48000
C++ How to Program, Dan Quirk,45000
Understanding Linux Programming,Bruce Molay,29000
Hands on Machine Learning,John Anderson,21000
Learning Python,Guido Van Rossum,27000
Effective Java,Joshua Bloch,32000
Machine Learning,Andreas Muller,3300
C# Programming,Hong Gil,45000
```

C Programming Final Exam(40 points)

■ Cautions

✓ Add your student number and name as comments at the top of the source file.

- If there are no comments, 1 point will be deducted per file.

✓ 0 points if a compile error occurs.

✓ If the execution result is hard coded, 0 points are given.

✓ If the conditions presented in the question are not followed, no points are given for that question.

✓ If you submit the exam source code late, 10% per minute will be deducted.

1. Program to compare the contents of two files and print line numbers (10 points)

- submission: final01.c

Write a program that compares two text files line by line and prints the line numbers and file contents for any differences. If file lengths differ, please check until the end of the file and print it to the screen.

- Files for use: sample1.txt, sample2.txt, sample3.txt

■ Same file comparison (2 points)

✓ Open two files in read mode and compare the file contents line by line.

✓ Do not print matching parts to the screen. If they are the same, print "The two files are identical." and exit the program.

✓ Even if the same file name is entered, a function to compare to the end of the file must be implemented: If there is no comparison functionality, zero points will be given.

■ Comparison of two files with different lengths and contents (8 points, each 4 points)

✓ When different parts are found, the actual file line number and file contents of each file are printed. (See example below)

- << 13: Objective-C, Objective-C++, Fortran, Ada, Go, D, Modula-2, Rust and COBOL among
- >> 13: Objective-C++, Fortran, Go, D, Modula-2, Rust and COBOL among others.[6]

✓ If the length of the two files is different, the line number and file contents of the longer file are printed.

■ Execution result #1: example of entering the same file name twice.

```
Type the first file name: sample1.txt
Type the second file name: sample1.txt
The two files are identical.
```

- Comparison of different files #2
  - ✓ Enter file names in various orders, such as sample2.txt, sample1.txt, etc.

- Execution result #2: sample2.txt, sample1.txt

```
Type the first file name: sample2.txt
Type the second file name: sample1.txt
<< 13: Objective-C++, Fortran, Go, D, Modula-2, Rust and COBOL among others.[6]
>> 13: Objective-C, Objective-C++, Fortran, Ada, Go, D, Modula-2, Rust and COBOL among
<< 14: The OpenMP and OpenACC specifications are also supported in the C and C++
>> 14: others.[6] The OpenMP and OpenACC specifications are also supported in the C and C++
>> 24:     --> sample1.txt 파일에만 존재하는 내용 출력
>> 25: GCC has been ported to more platforms and instruction set architectures than any
>> 26: other compiler, and is widely deployed as a tool in the development of both free and
>> 27: proprietary software. GCC is also available for many embedded systems, including
>> 28: ARM-based and Power ISA-based chips.
```

- Execution result #2: sample1.txt, sample3.txt

```
Type the first file name: sample1.txt
Type the second file name: sample3.txt
<<  1: The GNU Compiler Collection (GCC) (formerly GNU C Compiler) is a collection of
>>  1: ANSI C and ISO C
<<  2: compilers from the GNU Project that support various programming languages, hardware
>>  2: During the late 1970s and 1980s,
<<  3: architectures, and operating systems. The Free Software Foundation (FSF) distributes
>>  3: versions of C were implemented for a wide variety of mainframe computers,
<<  4: GCC as free software under the GNU General Public License (GNU GPL). GCC is a key
>>  4: minicomputers, and microcomputers, including the IBM PC,
<<  5: component of the GNU toolchain which is used for most projects related to GNU and
>>  5: as its popularity began to increase significantly.
<<  6: the Linux kernel. With roughly 15 million lines of code in 2019, GCC is one of the
>>  6: In 1983, the American National Standards Institute (ANSI) formed a committee,
<<  7: largest free programs in existence.[4] It has played an important role in the growth
>>  7: X3J11, to establish a standard specification of C.
<<  8: of free software, as both a tool and an example.
>>  8: X3J11 based the C standard on the Unix implementation;
<<  9:
>>  9: however, the non-portable portion of the Unix C library was handed off to the IEEE
<< 10: When it was first released in 1987 by Richard Stallman, GCC 1.0 was named the GNU C
>> 10: working group 1003
<< 11: Compiler since it only handled the C programming language.[1] It was extended to
>> 11: to become the basis for the 1988 POSIX standard.
<< 12: compile C++ in December of that year. Front ends were later developed for
>> 12: In 1989, the C standard was ratified as ANSI X3.159-1989 "Programming Language C".
<< 13: Objective-C, Objective-C++, Fortran, Ada, Go, D, Modula-2, Rust and COBOL among
>> 13: This version of the language is often referred to as ANSI C, Standard C, or
<< 14: others.[6] The OpenMP and OpenACC specifications are also supported in the C and C++
>> 14: sometimes C89.
<< 15: compilers.[7][8]
>> 15: In 1990, the ANSI C standard (with formatting changes) was adopted by the
```

```
<< 16:
>> 16: International Organization
<< 17: As well as being the official compiler of the GNU operating system, GCC has been
>> 17: for Standardization (ISO) as ISO/IEC 9899:1990, which is sometimes called C90.
<< 18: adopted as the standard compiler by many other modern Unix-like computer operating
>> 18: Therefore, the terms "C89" and "C90" refer to the same programming language.
<< 19: systems, including most Linux distributions. Most BSD family operating systems also
>> 19:
<< 20: switched to GCC shortly after its release, although since then, FreeBSD and Apple
>> 20: GCC was first released March 22, 1987, available by FTP from MIT.[18] Stallman was
<< 21: macOS have moved to the Clang compiler,[9] largely due to licensing
>> 21: listed as the author but cited others for their contributions, including Tower for
<< 22: reasons.[10][11][12] GCC can also compile code for Windows, Android, iOS, Solaris,
>> 22: "parts of the parser, RTL generator, RTL definitions, and of the Vax machine
<< 23: HP-UX, AIX, and MS-DOS compatible operating systems.[13]
>> 23: description", Jack Davidson and Christopher W. Fraser for the idea of using RTL as
<< 24:
>> 24: an intermediate language, and Paul Rubin for writing most of the preprocessor.[19]
<< 25: GCC has been ported to more platforms and instruction set architectures than any
>> 25: Described as the "first free software hit" by Peter H. Salus, the GNU compiler
<< 26: other compiler, and is widely deployed as a tool in the development of both free and
>> 26: arrived just at the time when Sun Microsystems was unbundling its development tools
<< 27: proprietary software. GCC is also available for many embedded systems, including
>> 27: from its operating system, selling them separately at a higher combined price than
<< 28: ARM-based and Power ISA-based chips.
>> 28: the previous bundle, which led many of Sun's users to buy or download GCC instead of
>> 29: the vendor's tools.[20] While Stallman considered GNU Emacs as his main project, by
>> 30: 1990 GCC supported thirteen computer architectures, was outperforming several vendor
>> 31: compilers, and was used commercially by several companies.[21]
```

2. Book management program using file input/output (30 points)

- submission: final02.c

Implement a program that reads the given file (books.txt), stores its contents in an array of structures, and then adds, searches, updates, and saves the final book information.

■ structure information

```c
#define MAX_BOOK 10

typedef struct {
    char title[50];
    char author[30];
    int price;
} book;

// global variable
int book_count = 0;  // the number of books stored in the structure array
int main()
{
    book booklib[MAX_BOOK] = {0};
    . . .
    return 0;
}
```

- ✓ The booklib[MAX_BOOK] structure array must be declared in the main() function and used as a pointer passed to each function (20 points will be deducted if used as a global variable)

■ **Implement a function that reads "books.txt" file and save it in a structure array(booklib[])** (2 points)

- ✓ Read the file in read mode and store it in the structure array
  - Each item in the file is separated by a comma(,).
  - Use the sample code below(strtok() can also be used)
  - Read the file and print the number of structures saved
  - example: Read books.txt (8 books) successfully.

```
char line[100] = {0};
. . .
// read one line at a time until the end of the file and store it in booklib[]
fgets(line, sizeof(line), fin);
sscanf(line, "%[^,],%[^,],%d",
         booklib[index].title,
         booklib[index].author,
         &booklib[index].price);
```

■ Each menu is implemented with separated functions (28 points)

- ✓ 1. Implementing a function for "Print All Books" menu **(4 points)**
  - Print all items in the structure array
  - Refer to the execution result.
- ✓ **2. Implementing a function for "Add New Book" menu (4 points)**
  - Up to 10 book information can be added to the structure array
  - Title and author must be able to store spaces in the middle of the string
  - After adding book information, the current data count is displayed. (See execution results)
  - If 10 data are saved and no more are saved, the screen will display "No empty space. (10 Books)"
  - After saving new book information, the saved data must be printed through menu 1.

- ✓ **3. Implementing a function for “Search Book” menu** (8 points, No partial point)
  - Search conditions: display all books whose title field contains the entered keyword and whose price value is within the specified price range.
  - Print the total search count
  - If there is no title containing the keyword or no book in the price range, print “No Search Result(keyword, price range)”
- ✓ **4. Implementing a function for Update Book Information** (10 points)
  - Enter the index to change (1~book_count)
  - If the index exceeds the range of the array, the function exits after printing “Out of index”.
  - If the index is within the normal range, output the data for that index. (1 point)
  - If you only press Enter in the title and author data to be changed (4 points)
    - The title and author fields retain existing data.
    - Price field must be entered.
  - If new data is entered for the title, author, or price to be changed (4 points)
    - Change data to the entered title, author, and price.
  - Print changed data (1 point)
- ✓ **5. Save Book Data and Exit Program** (2 points)
  - Save the entire contents of the currently changed structure array to “book_out.txt” file.
    - Print the total number of data saved as a file.
    - Check “book_out.txt” in VS code.

■ Execution result

```
Read books.txt (8 books) successfully.  ←---- Read “books.txt” first
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 1
Total Book Count: 8
[1] Title: C Programming, Author: Dennis Ritchie, Price: 18000
[2] Title: Modern C Programming, Author: K.N.King, Price: 35000
[3] Title: Java How to Program, Author:  Paul Deitel, Price: 48000
[4] Title: C++ How to Program, Author:  Dan Quirk, Price: 45000
```

```
[5] Title: Understanding Linux Programming, Author: Bruce Molay, Price: 29000
[6] Title: Hands on Machine Learning, Author: John Anderson, Price: 21000
[7] Title: Learning Python, Author: Guido Van Rossum, Price: 27000
[8] Title: Effective Java, Author: Joshua Bloch, Price: 32000

====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 2
Type a new book title(Max: 50 chars):  Machine Learning
Type a new book author(Max: 30 chars): Andreas Muller
Type a new book price: 3300
A new book added.(book count: 9)  ←---- Print the number of books stored in the array
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 2
Type a new book title(Max: 50 chars): Linux Programming
Type a new book author(Max: 30 chars): Neil Matthew
Type a new book price: 39000
A new book added.(book count: 10)
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 2
No empty space.(10 Books)    ←------ 10 data are stored, no more can be added.
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
```

```
====================================
Type a number: 1               ←-------- Print all saved data.
Total Book Count: 10
[1] Title: C Programming, Author: Dennis Ritchie, Price: 18000
[2] Title: Modern C Programming, Author: K.N.King, Price: 35000
[3] Title: Java How to Program, Author:  Paul Deitel, Price: 48000
[4] Title: C++ How to Program, Author:  Dan Quirk, Price: 45000
[5] Title: Understanding Linux Programming, Author: Bruce Molay, Price: 29000
[6] Title: Hands on Machine Learning, Author: John Anderson, Price: 21000
[7] Title: Learning Python, Author: Guido Van Rossum, Price: 27000
[8] Title: Effective Java, Author: Joshua Bloch, Price: 32000
[9] Title: Machine Learning, Author: Andreas Muller, Price: 3300
[10] Title: Linux Programming, Author: Neil Matthew, Price: 39000
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 3
Type a title keyword for search: Program          ←--Include “Program” in the title
Type a price range(ex: 10000 20000): 15000 35000  ←-- search for books with the price
[1] Title: C Programming, Author: Dennis Ritchie, Price: 18000                  range
[2] Title: Modern C Programming, Author: K.N.King, Price: 35000
[5] Title: Understanding Linux Programming, Author: Bruce Molay, Price: 29000
Total 3 books found


====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 3
Type a title keyword for search: Java
Type a price range(ex: 10000 20000): 10000 20000
No Search Result(Java, 10000 ~ 20000) ←----- If there are no books in the price range


====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 3
Type a title keyword for search: AI
Type a price range(ex: 10000 20000): 10000 50000
No Search Result(AI, 10000 ~ 50000) ←----- No title containing “AI”
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 4
Type a index for updating: 11      ←----- Entered out of index range(1~book_count)
Out of index
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 4
Type a index for updating: 0        ←----- Entered out of index range(1~book_count)
Out of index
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 4
Type a index for updating: 1
[1] Title: C Programming, Author: Dennis Ritchie, Price: 18000  ←- print the index data
Enter a new title(Enter: skip):          ←---- Typed Enter key in title field
Enter a new author name(Enter: skip):    ←---- Typed Enter key in author field
Enter a new book price: 1800
Updated Successfully                      ←------- Only the price has changed.
[1] Title: C Programming, Author: Dennis Ritchie, Price: 1800 ←-- Print changes.
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 4
Type a index for updating: 10
[10] Title: Linux Programming, Author: Neil Matthew, Price: 39000
Enter a new title(Enter: skip):  C# Programming
Enter a new author name(Enter: skip): Hong Gil
Enter a new book price: 45000
Updated Successfully
[10] Title: C# Programming, Author: Hong Gil, Price: 45000
====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 1
Total Book Count: 10
[1] Title: C Programming, Author: Dennis Ritchie, Price: 1800
[2] Title: Modern C Programming, Author: K.N.King, Price: 35000
[3] Title: Java How to Program, Author:  Paul Deitel, Price: 48000
[4] Title: C++ How to Program, Author:  Dan Quirk, Price: 45000
[5] Title: Understanding Linux Programming, Author: Bruce Molay, Price: 29000
[6] Title: Hands on Machine Learning, Author: John Anderson, Price: 21000
[7] Title: Learning Python, Author: Guido Van Rossum, Price: 27000
[8] Title: Effective Java, Author: Joshua Bloch, Price: 32000
[9] Title: Machine Learning, Author: Andreas Muller, Price: 3300
[10] Title: C# Programming, Author: Hong Gil, Price: 45000

====================================
 Book Management Program
 1. Print All Books
 2. Add New Book
 3. Search Book
 4. Update Book Information
 5. Save Book Data and Exit Program
====================================
Type a number: 5
10 Book Data Saved.
Exit Program
```

■ File contents of “book_out.txt”

```
C Programming,Dennis Ritchie,1800
Modern C Programming,K.N.King,35000
Java How to Program, Paul Deitel,48000
C++ How to Program,Dan Quirk,45000
Understanding Linux Programming,Bruce Molay,29000
Hands on Machine Learning,John Anderson,21000
Learning Python,Guido Van Rossum,27000
Effective Java,Joshua Bloch,32000
Machine Learning,Andreas Muller,3300
C# Programming,Hong Gil,45000
```