

940 Blanshard Street  
Victoria, British Columbia  
V8W 3E6  
Susan Fiddler  
Co-op Coordinator  
Faculty of Engineering  
University of Victoria  
P.O. Box 1700  
Victoria, B.C.  
V8W 2Y2

Dear Susan,

Please accept the accompanying Work Term Report entitled "Determining uses of JIRA and Confluence at MOTI IMB."

This report is the result of work completed at the SOME GENERiC ORGANIZATION. During my first work

term as a University of Victoria student, I used charts and tables to display information about issue, complied documentation for critical applications in a wiki and researched additions to extend functionality. In the course of work, I gained exposure to a technical environment, and learned how software can integrate together.

Through the course of the term, I was given the opportunity to learn much agile software development, testing applications, and software products. I feel that this knowledge will be helpful in future work terms, and in my career.

I would like to thank my manager, MISTER MAN, for his patience and good judgement, as well as the RANDOM FOLK who were always willing to help.

Sincerely,

David Li

# Table of Contents

List of Figures . . . . .	iii
List of Tables . . . . .	iii
Summary . . . . .	iv
<b>1 Introduction . . . . .</b>	<b>1</b>
<b>2 Discussion . . . . .</b>	<b>2</b>
2.1 JIRA . . . . .	2
2.1.1 Overview . . . . .	2
2.1.2 Issues . . . . .	3
2.1.3 Dashboards . . . . .	4
2.1.4 Agile Projects in JIRA . . . . .	5
2.1.5 Add-ons . . . . .	6
2.1.6 Limitations . . . . .	8
2.2 Confluence . . . . .	8
2.2.1 Overview . . . . .	9
2.2.2 Spaces . . . . .	9
2.2.3 Pages . . . . .	9
2.2.4 Embedding Content . . . . .	10
2.2.5 Limitations . . . . .	10
2.3 JIRA and Confluence Integration . . . . .	10
<b>3 Conclusion . . . . .</b>	<b>11</b>
<b>4 Recommendations . . . . .</b>	<b>12</b>
<b>Appendix A Code Listings . . . . .</b>	<b>12</b>

## List of Figures

1	The Block . . . . .	1
2	Clouds . . . . .	3
3	People . . . . .	4
4	Visuallizing Data . . . . .	5
5	Typical Pictures . . . . .	6
6	Final Pictures . . . . .	7
7	Cool Looking USBs. . . . .	8
8	Creating Life . . . . .	9
9	Good Picture . . . . .	10
10	Good Looking Report . . . . .	11

## List of Tables

1	Time log for ELEC 360 — Assignment 2 . . . . .	6
---	--	---

## Summary

In the continuing effort to organize high-quality reliable information, the **INSERT PREVIOUS EMPLOYEE INFORMATION DIVISION** is presently experimenting with **JIRA**, an issue tracking tool and **Confluence**, wiki software for technical documentation. Although good quality information is critical to operations, **documentation** is inconsistent and scattered across multiple sources, some of which require access permissions. **JIRA** and **Confluence** are software tools that improve productivity and organization within MOTI IMB.

Benefits from connecting **JIRA** and **Confluence** include common user management, reporting on existing **JIRA issues** in **Confluence** and switching between application quickly. Extending the functionality of these tools by installing add-ons will assist in improving **documentation**. Purchasing software such as **Confluence** to solve the **documentation** problem is inadequate because software can be poorly designed or implemented. These software tools assist in information management, but full utilization and proper implementation is required to improve documentation.

# 1. Introduction

The **MOTI** (MOTI) is responsible for transportation services, infrastructure and formulates transportation policies. The **MOTI IMB** strives to empower the **MOTI** to achieve effective use of and greatest benefit from information technology management.



Figure 1: The Block

Atlassian, an enterprise software company, produces software tools including **JIRA** and **Confluence** for team collaboration, software development and organization. At MOTI IMB, enterprise software is used for issue tracking, project management and organize content. Currently, information is scattered across LAN DRIVES, **Microsoft SharePoint** Sites and other sources that work for small groups of people. Going forward, using enterprise software such as JIRA and Confluence is essential to manage projects and their issues throughout the software development lifecycle.

Inspecting the capabilities of JIRA and Confluence is helpful to streamlining the adoption of these versatile software products. While JIRA offers flexible issue tracking and project management, Confluence is wiki software used for technical documentation that locates information quickly and easily. Improving and standardizing documentation for applications requires methodically examining software capabilities. Criteria include functionality, compatibility, ease of use, strategic purpose and integration with existing software. Switching to JIRA has reduced email volume, improved customer support and simplified project management. Currently, the use Confluence restricted to selected individuals.

Analysing the potential uses of Confluence and JIRA will provide insight on how to better use these new tools.

## 2. Discussion

Oftentimes, tacit information is trapped within emails, but using **JIRA** and **Confluence** together will capture this information forever. In addition, **Confluence** and **JIRA** have permissions to give flexibility to decide who can view and edit content.

### 2.1. JIRA

Before JIRA Microsoft SharePoint, a document management system was used to store, organize, share, and access information [1]. An issue in JIRA is used to create, track and resolve reported client issues. MOTI IMB employees use JIRA on a daily basis, so an issue could represent a software bug, a project task, a helpdesk ticket, etc. [2].

JIRA is highly customizable, organized by projects, components, versions and labels. The most flexible way to search for issues is by constructing JQL (JIRA Query Languages) filters that use issue fields, operators, values and keywords and return specified JIRA issues. JIRA has split into 3 different standalone products JIRA Core, JIRA software, and JIRA Service Desk. Designed for managing projects and tasks, JIRA Core keeps team organized. JIRA Software allows for Agile boards and integrates with software development tools. JIRA Service Desk is IT service desk and customer service software. JIRA incorporates time tracking, reporting features and is enhanced by installing add-ons such as ScriptRunner and Tempo Timesheets. Despite regular use of JIRA at MOTI IMB, its extensive functionality and plethora of features will confuse newcomers.

#### 2.1.1. Overview

Although JIRA originated as a software tool geared towards developers, it has evolved to address the needs of the non-technical users. **JIRA** can track issues, display information on a dashboard and manage **agile** projects by using **Kanban** or **Scrum** Boards which are designed to manage software projects. JIRA permissions are setting that limit what users can see and do. JIRAs permission scheme includes editing issues, creating issues, commenting, assigning issues, etc. [3]. Generally, users are added to an JIRA project, begin searching for issues, close **issue** and repeat.

### 2.1.2. Issues

An **issue** contains multiple fields including type, priority, component, status, resolution, description assignee and reporter. It contains information indicating the urgency and current status of the issue. Some of the options available for issue status field include new, to-do, in progress, hold, and closed. Issue priorities vary from trivial to blocker. Classifying issues correctly is essential for users to document their work and reference in the future. Issues can have subtasks, be duplicated or be related to other existing issues. Watching an issue notified the user when the issue is commented, status changed, closed, or updated.



Figure 2: Clouds

When the status of an issue is changed or closed the reporter will receive an email update. Time tracking in **JIRA** is accomplished by creating a work log, entering time spent, giving a reasonable remaining estimate and description of work done. Oftentimes, issues are reassigned to assignees better suited to complete the task. Attachments can be added to the issue to better describe work that needs to be done. Previously at IMB, for each application a Microsoft **SharePoint** site would be created, and issue tracking was managed by creating list items. Even though this worked for small groups, providing access to each individual SharePoint site was cumbersome and time-consuming for everybody. A single system that contains issues for all projects benefits all users and has become essential to day-to-day operations.



Issues in JIRA can be imported or exported to and from xls files, which allows the user to reuse content from fabricated in other applications. Overall, JIRA issues can represent anything ranging from software bugs, to an access request and used for project management.

### 2.1.3. Dashboards

JIRA dashboards contains information boxes known as gadgets that can display dynamic data about a JIRA filter or project. Gadgets let you customize the information that appears on dashboards. Configuring dashboards allows different kinds of information to be displayed. Most of these gadgets require JIRA filters to feed information into it. Dashboards can display filter results, issue statistics, pie charts grouped by a specified issue field such as created, assignee, status, and priority. Add-ons for JIRA should as Zephyr (used for test management) and Tempo Timesheets (creates timesheets for JIRA issues) are packaged with dashboard gadgets containing additional information specific to that add on. An example is the user timesheet add-on from tempo which displays a timesheet for a specific period of time.



Figure 3: People

At this time, dashboard usage is inconsistent, and undergoing further investigation. Ideally, every project and every team would have a JIRA dashboard, customized to suit their



individual needs.

Also, **JIRA** has adaptable **Kanban** and **Scrum** boards to track issue backlogs. These boards can be displayed on a JIRA dashboard. Creating a meaningful dashboard faces several obstacles including understanding of the JIRA specific query language, **JIRA Query Language (JQL)**, knowledge of which dashboard gadget is suitable and a steep learning curve.

Developing an informative **JIRA** dashboard entails consulting with the end users, multiple iterations and producing JIRA filters.



Figure 4: Visualizing Data

Dashboards are the best method to visually reporting on ongoing and completed issues in **JIRA**.

#### 2.1.4. Agile Projects in JIRA

Agile software development is used by the PEOPLE to produce software that better suits the needs of customers. In **agile**, requirements and solutions develop through frequent

Table 1: Time log for ELEC 360 — Assignment 2			
Week of Oct 8		Week of Oct 15	
5 hours Prepared for ELEC 360 Lab and completed it		2 hours working on lab report	3 hours spent
3 hours of studying for quizzes		3 hours studying for midterm	2 hours studying
1 hour creating notes		2 hours preparing for midterm	2 hours preparing

conversations with stakeholders as software is developed through collaboration. Two kinds of agile boards are available in JIRA Agile, Kanban and Scrum. Each of these boards are designed to follow their specific agile methodology. Adding the fix version field to an issue will assign it to a particular release. All released and unreleased versions can be viewed within the project sidebar in JIRA. JIRA software provides tools specifically for any agile project management which include boards, releases and dashboard gadgets (burndown charts, velocity charts and more).



Figure 5: Typical Pictures

Altogether, JIRA is well-suited for **agile** project management with boards, versioning and dashboard gadgets.

### 2.1.5. Add-ons

JIRA add-ons provide additional functionality to JIRA including increased customization and features.

Adaptavist ScriptRunner provides custom workflows, custom JQL and executes scripts that interact with JIRA. Some notable features include custom automatic emails triggered when an issue transitions into a particular state, enhanced JIRA filters (i.e. searching for issues attachments) and adding additional fields for JIRA issues. This add-on provides a simple way to script in JIRA.

Tempo timesheets enhances time tracking and reporting features while faultlessly integrating with JIRA. It helps teams and managers track time, be efficient and used for forecasting. This add-on is useful because it allows JIRA users to accurately track their time spent on issues, projects and make strategic decisions based on this info.

The tempo time tracker runs within JIRA and is the best way to track time.



Figure 6: Final Pictures

A user timesheet contains information about hours worked, planned, and required for the week. It organizes issues automatically organizes issues by project and component.



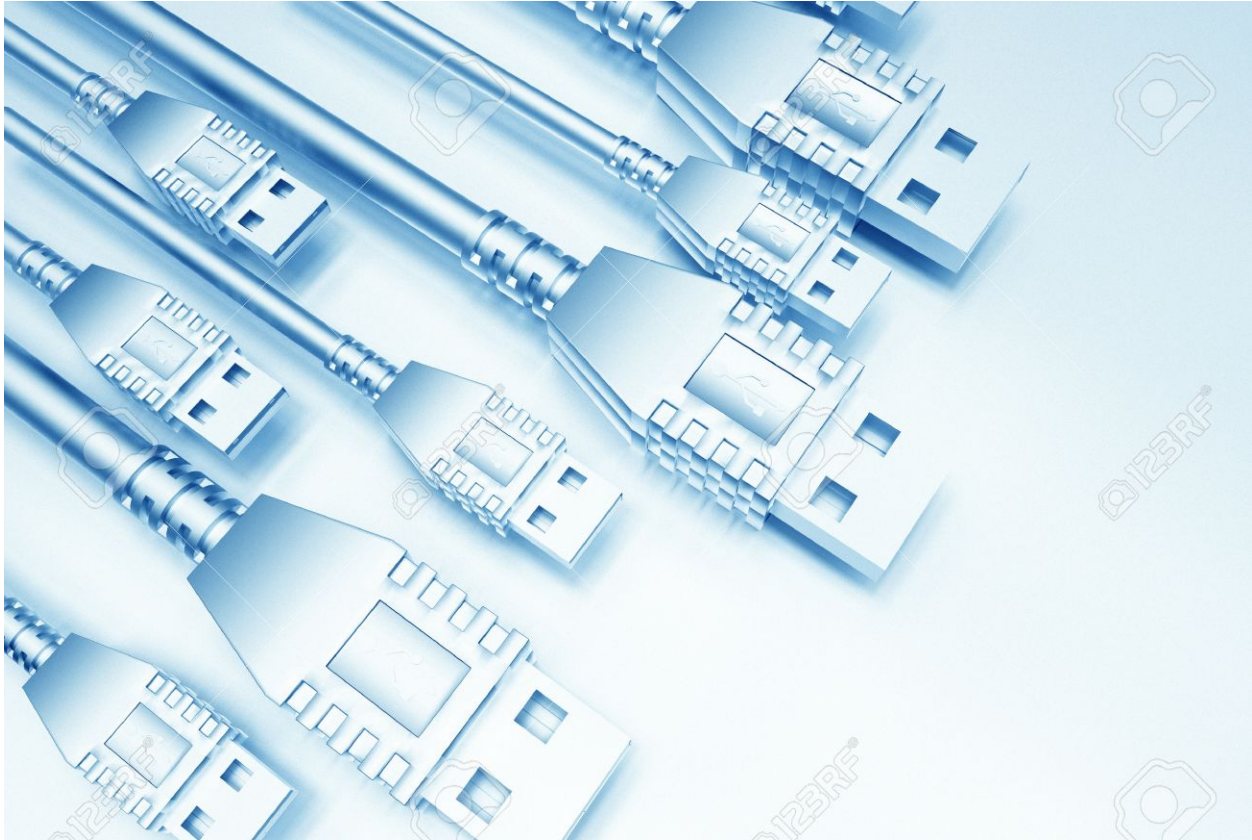


Figure 7: Cool Looking USBs.

### 2.1.6. Limitations

JIRA can be extremely slow at times which may be caused by running out of memory, and a large JIRA instance [4]. Extensive functionality such as time tracking, **agile** boards, dashboards and user timesheets are underutilised because users unaccustomed to JIRA. Additionally, a lack of tutorials or instructions are how to use JIRA compound the difficult of learning how to use **JIRA**.

## 2.2. Confluence

**Confluence** is team-based collaboration software developed by Atlassian used at the ORGANIZATION to document projects and work in conjunction with **JIRA**. Confluence works by It organizes content by labels, can embed videos on pages and is highly customizable. The aim of Confluence is to become the tool for organizing, discussing, and doing work. Assigning tasks, recording meeting notes and sharing documents are some of the features of Confluence.

### 2.2.1. Overview

Confluence keeps page and file versions to track every version of documents. Despite numerous free wiki-software available on the web, Confluence has page hierarchy (page trees), transparency when creating content and clean user interface which make it suitable for documentation.

### 2.2.2. Spaces

Everything in Confluence is organized in spaces, which are a collection of related pages. Pages are the documents in which the team will create, edit, and discuss work. Confluence Spaces are organized by categories (i.e., critical non-critical, documentation and software-project). Spaces can be customized to fit the need of each team. For example uses of Confluence include knowledge bases, technical documentation and organizing content [5].

### 2.2.3. Pages

In a Confluence page, content is created, collected, and accessed in one place.

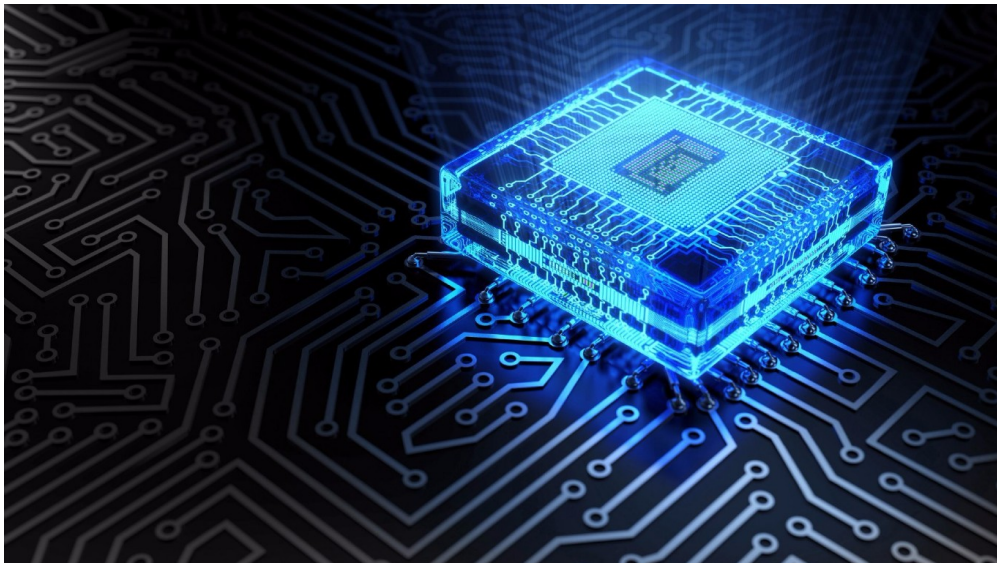


Figure 8: Creating Life

**Confluence** pages contain text, images, videos and uses macros that add extra functionality or include dynamic content. Macros can create charts, incorporate existing Microsoft documents (Word, Excel, PowerPoint) and reuse content on Confluence pages. Existing templates in Confluence serve as an excellent starting point for users to start creating documentation for Confluence. Customizing page and space templates is an essential feature to save time and reuse content.

### 2.2.4. Embedding Content

Attachments are useful for sharing existing content saved in a different file format.



Figure 9: Good Picture

When word documents imported into Confluence, that document's content is copied onto one or more Confluence pages. Additionally, images and videos from the web can be embedded into a Confluence Page. Sharing links is convenient and helps users find information.

### 2.2.5. Limitations

Confluence is designed for documentation, used by large organizations including NASA [6], yet documenting legacy and current projects in Confluence is not occurring at this moment. Currently, use of Confluence is limited to few internal users, so education about this software and its functionality lacking. Certain files types such as mpp (Microsoft Project) and vsd (Microsoft Visio) cannot be embedded within a Confluence Page, but can be uploaded as a file. Creating structured space templates and pages will allow content to be streamlined. Confluence functionality can be extending by downloading add-on in the Atlassian marketplace. For example, tables in Confluence are rudimentary, consisting only of ascending/descending sorting, however, a few add-ons for Confluence address this problem.

## 2.3. JIRA and Confluence Integration

Much like how Microsoft Office Products integrate well together, JIRA and Confluence are better together. When **Confluence** and JIRA sites are connected JIRA issues can be created, viewed and more from within Confluence. Also, connecting JIRA with Confluence allows pie charts, created vs resolved chart and filter results etc. Essentially, existing information stored in JIRA can be reported in Confluence by the JIRA issues macro or the JIRA charts macro [7].



Another benefit from connecting JIRA to **Confluence** is delegated user management to **JIRA**, so all users can be managed in a single location. Whenever a link is created to a JIRA issue in Confluence, or link to a Confluence page from a JIRA application, the JIRA Links button appears at the top of the Confluence page. This makes it very simple to jump from JIRA straight into Confluence and vice-versa.

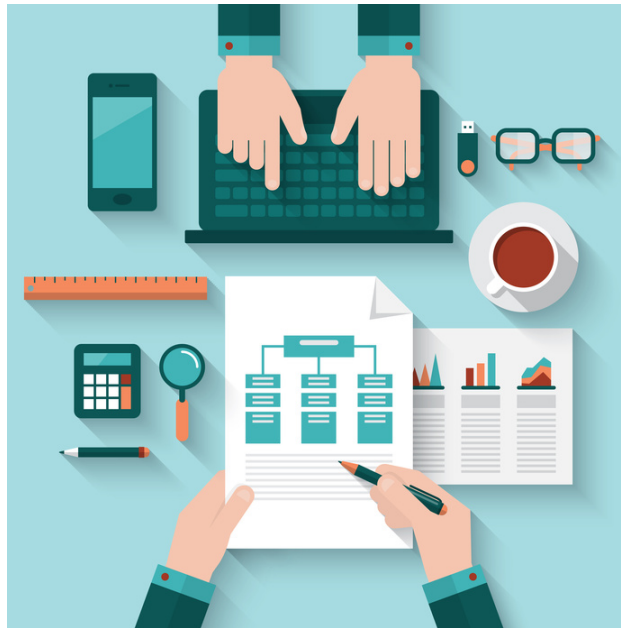


Figure 10: Good Looking Report

The tight integration between Confluence and JIRA results in related issues can be accessed directly from the Confluence page with their current status.

### 3. Conclusion

Adopting JIRA has benefited ME by reducing response time, improving record keeping and sharpening communication. Although JIRA has benefited the organization by improving efficiency, lack of education resulting in underutilisation of dashboards and filters, occasional performance issues causing issue backlogs to pile up, and perceived unreliability for sending emails. Agile software development methodologies such as Scrum and Kanban are supported by the JIRA suite of applications. JIRA has add-ons that extend functionality will complexity ranging from a single feature to full product. Interestingly, when JIRA is integrated with Confluence usage of both applications will increase because jumping between application is quick and simple.

In order to streamline the adoption of **Confluence**, usage intentions must be conveyed as well as detailed guidance on how to use it. Confluence is wiki software used to create

knowledge bases, centralize information into a single system and for technical documentation. Useful features of Confluence include integration with **JIRA**, word processing, and team collaboration. Connecting JIRA and Confluence applications allows enable users to rapidly switch between applications. Overall, JIRA and Confluence are software tools that improve productivity and organization within BTI IM.

## 4. Recommendations

Even though JIRA and Confluence are powerful software tools, failing to grasp the full capabilities limits what can be done. Potential uses of JIRA include issue management, project management and lengthy daily activities that require documentation. Although there are multiple issues types at this time, clear definitions for these issues and when to use them are missing. Adding a wider range of types will allow JIRA to be for a wider range of problems. Also, users are inconsistent in their use of JIRA, failing to include details, documents or work offline. Increasing understanding of how it works and guidance on when to use it needed. Creating business rules on how and when to use JIRA and Confluence will improve documentation consistency. Confluence should be configured to support day to day JIRA activities at IBF MSD. Furthermore, research into the plethora of add-ons available for JIRA and Confluence to see what functionality can be added.

## 5. References

- [1] *What is SharePoint? - Office Support*. [Online] Available: <https://support.office.com/en-us/article/What-is-SharePoint-97b915e6-651b-43b2-827d-fb25777f446f>. Accessed December 11, 2016.
- [2] *What is an Issue*. [Online] Available: <https://confluence.atlassian.com/jira064/what-is-an-issue-720416138.html>. Accessed December 11, 2016.
- [3] *Permissions overview*. [Online] Available: <https://confluence.atlassian.com/jirasoftwarecloud/permissions-overview-764478244.html>. Accessed December 13, 2016.
- [4] *Crashes and Performance Issues Troubleshooting*. [Online] Available: <https://confluence.atlassian.com/jirakb/crashes-and-performance-issues-troubleshooting-203394749.html>. Accessed December 15, 2016.
- [5] *Advanced and Special Uses of Confluence*. [Online] Available: <https://confluence.atlassian.com/doc/integrating-jira-and-confluence-2825.html>. Accessed December 17, 2016.
- [6] *Confluence at NASA: Where No Wiki Has Gone Before*. [Online] Available: <http://summit.atlassian.com/archives/2010/presentations/collaboration-and-projects/confluence-wiki-at-nasa.jsp>. Accessed December 13, 2016.

- [7] *Integrating JIRA and Confluence*. [Online] Available: <https://confluence.atlassian.com/doc/integrating-jira-and-confluence-2825.html>. Accessed December 17, 2016.

## A. Code Listings

Listing 1: Caesar Cipher for CSC 111

```
/*
 * Author:    David Li
 * UVicID:    V00818631
 * Date:      Oct 69, 2014
 * Lecture:   Assignment 7
 * File name: V00818631A7P3.c
 * Description: Reading, writing, and encoding files
 */

#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <ctype.h>
#include <string.h>
#include <time.h>

#define INPUTFILENAME ("A7_2014_P3_TestingSherlock.txt")
#define OUTPUTFILENAME ("A7_2014_P3_SherlockEncoded.txt")
#define MAXSWAP (4)

void processFile(FILE* ifp, FILE* ofp, int seed) {
    char line[25];
    char word[25];
    printf("Begin process file\n");
    // your code goes here
    int rn1, rn2;
    int j;
    char tmp;

    while(!feof(ifp)) {

        fscanf(ifp, "%s", line);
        strcpy(word, line);
```

```

        if(strlen(line) > 3) {
            for(j=0; j<MAXSWAP; j++) {
                rn1 = rand() % (strlen(line) -2);
                rn2 = rand() % (strlen(line) -2);
                tmp = line[rn1 + 1];
                line[rn1 + 1] = line[rn2 + 1];
                line[rn2 + 1] = tmp;
            }
            if(strcmp(word,line) == 0) {
                tmp = line[1];
                line[1] = line[2];
                line[2] = tmp;
            }
        }

        if(!feof(ifp))
            fprintf(ofp,"%s ",line);

    }
    printf("End process file\n");
} /* ProcessFile */


int main(void) {
    printf("Welcome to Sherlock Holmes\n\n");
    unsigned int seed = (unsigned int)time(NULL);
    srand(seed);
    FILE *ifp;
    FILE *ofp;
    ifp = fopen(INPUTFILENAME, "r");
    if (ifp == NULL) {
        printf("Cannot open input file %s\n", INPUTFILENAME);
        exit(EXIT_FAILURE);
    } /*if*/
    ofp = fopen(OUTPUTFILENAME, "w");
    if (ofp == NULL) {
        printf("Cannot create output file %s\n", OUTPUTFILENAME);
        exit(EXIT_FAILURE);
    } /*if*/
    processFile(ifp, ofp,seed);
    fclose(ofp);
    fclose(ifp);
    printf("\nWe encoded Sherlock Holmes\n");
    return EXIT_SUCCESS;
}

```

```
} /*main*/
```

---