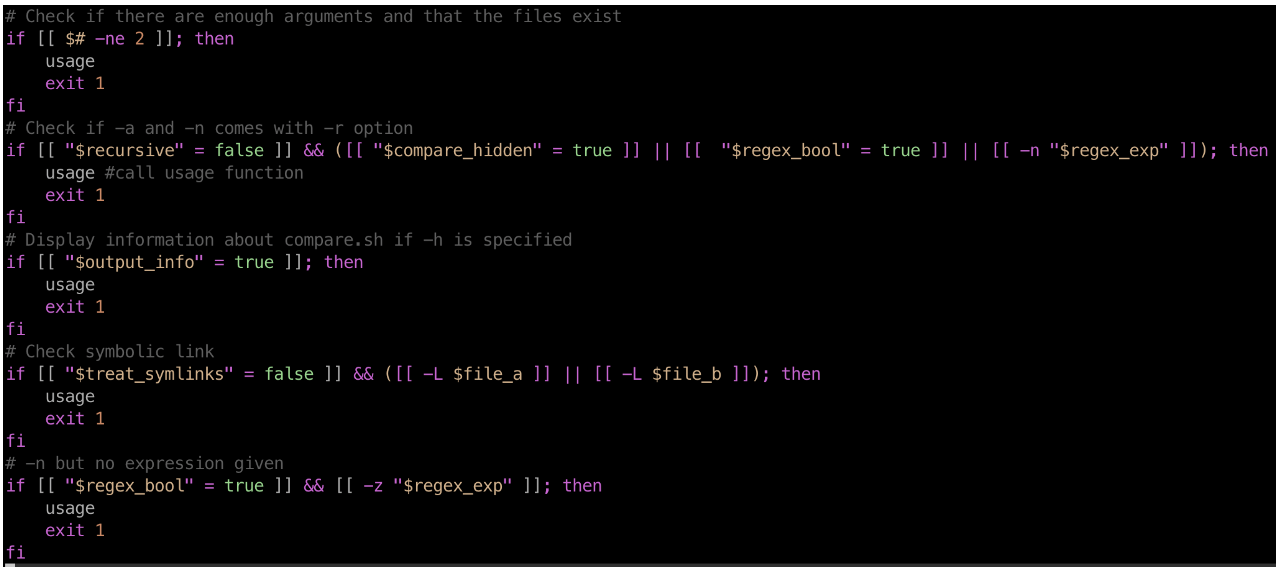
Author : IM B12705014 陳泊華

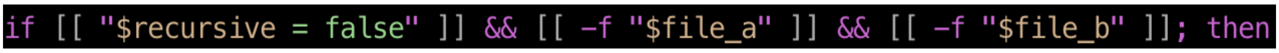
NASA hw2

1 參數檢查

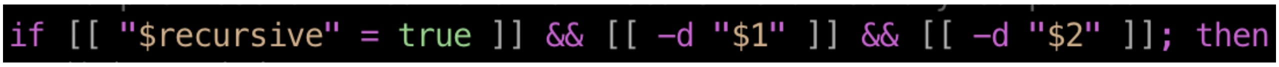


```
# Check if there are enough arguments and that the files exist
if [[ $# -ne 2 ]]; then
    usage
    exit 1
fi
# Check if -a and -n comes with -r option
if [[ "$recursive" = false ]] && ([[ "$compare_hidden" = true ]] || [[  "$regex_bool" = true ]] || [[ -n "$regex_exp" ]]); then
    usage #call usage function
    exit 1
fi
# Display information about compare.sh if -h is specified
if [[ "$output_info" = true ]]; then
    usage
    exit 1
fi
# Check symbolic link
if [[ "$treat_symlinks" = false ]] && ([[ -L $file_a ]] || [[ -L $file_b ]]); then
    usage
    exit 1
fi
# -n but no expression given
if [[ "$regex_bool" = true ]] && [[ -z "$regex_exp" ]]; then
    usage
    exit 1
fi
```

想法：如上圖，逐一檢視題目給的參數條件進行判斷式的處理；其中判斷檔案/目錄是否合法的部分寫在下面進行compare的區塊



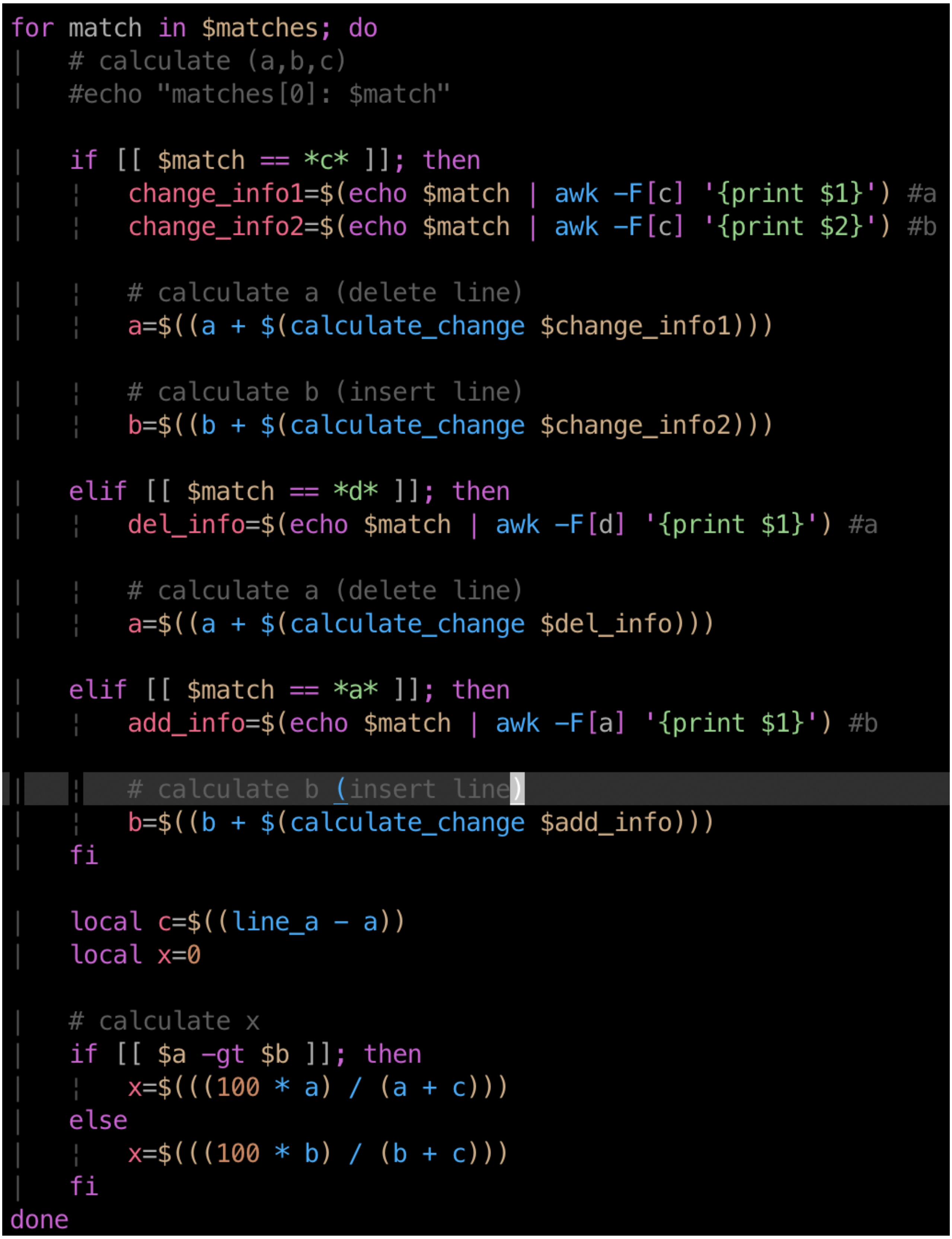
```
if [[ "$recursive = false" ]] && [[ -f "$file_a" ]] && [[ -f "$file_b" ]]; then
```



```
if [[ "$recursive" = true ]] && [[ -d "$1" ]] && [[ -d "$2" ]]; then
```

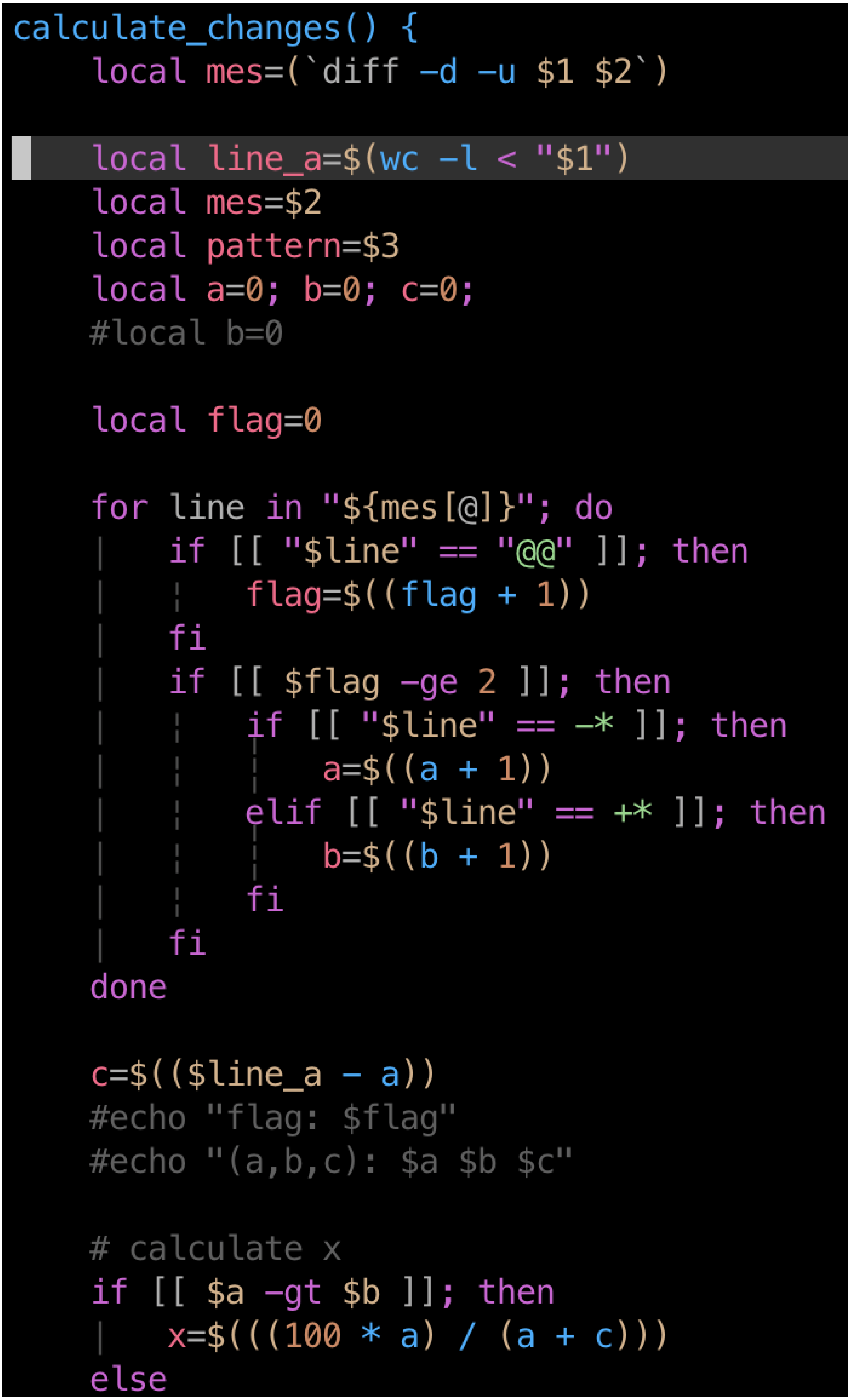
2 比較檔案

想法：嘗試過直接parse `diff -d` 的輸出（如下圖），主要邏輯為使用awk等指令篩選出如"1c1", "3,5c4,5"之類的資訊進行運算：

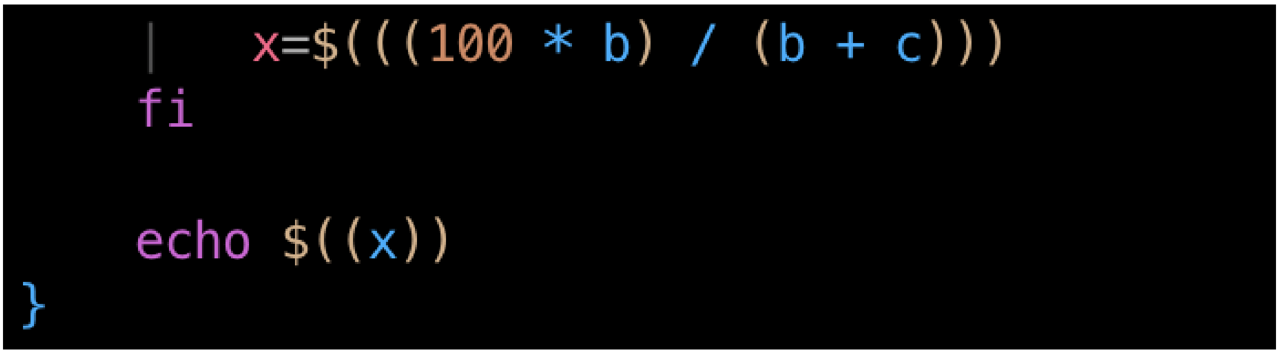


```
for match in $matches; do
    # calculate (a,b,c)
    #echo "matches[0]: $match"

    if [[ $match == *c* ]]; then
        change_info1=$(echo $match | awk -F[c] '{print $1}') #a
        change_info2=$(echo $match | awk -F[c] '{print $2}') #b

        # calculate a (delete line)
        a=$((a + $(calculate_change $change_info1)))

        # calculate b (insert line)
        b=$((b + $(calculate_change $change_info2)))

    elif [[ $match == *d* ]]; then
        del_info=$(echo $match | awk -F[d] '{print $1}') #a

        # calculate a (delete line)
        a=$((a + $(calculate_change $del_info)))

    elif [[ $match == *a* ]]; then
        add_info=$(echo $match | awk -F[a] '{print $1}') #b

        # calculate b (insert line)
        b=$((b + $(calculate_change $add_info)))
    fi

    local c=$((line_a - a))
    local x=0

    # calculate x
    if [[ $a -gt $b ]]; then
        x=$(((100 * a) / (a + c)))
    else
        x=$(((100 * b) / (b + c)))
    fi
done
```

後來發現 `diff -d -u` 的參數，於是進行簡化，程式碼如下：



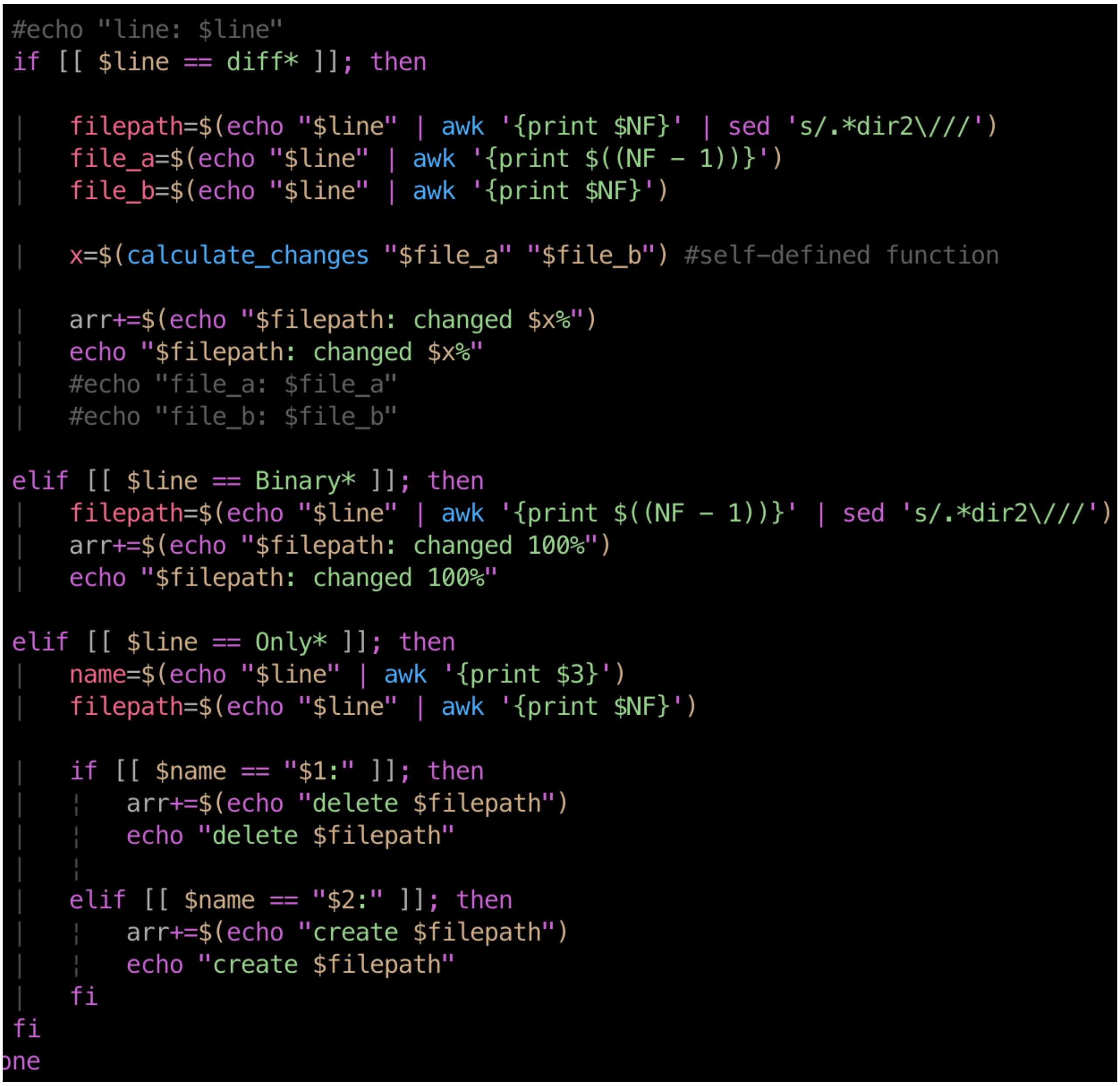
```
calculate_changes() {
    local mes=(`diff -d -u $1 $2`)

    local line_a=$(wc -l < "$1")
    local mes=$2
    local pattern=$3
    local a=0; b=0; c=0;
    #local b=0

    local flag=0

    for line in "${mes[@]}"; do
        if [[ "$line" == "@@" ]]; then
            flag=$((flag + 1))
        fi
        if [[ $flag -ge 2 ]]; then
            if [[ "$line" == -* ]]; then
                a=$((a + 1))
            elif [[ "$line" == +* ]]; then
                b=$((b + 1))
            fi
        fi
    done

    c=$(($line_a - a))
    #echo "flag: $flag"
    #echo "(a,b,c): $a $b $c"

    # calculate x
    if [[ $a -gt $b ]]; then
        x=$(((100 * a) / (a + c)))
    else
```



```
        x=$(((100 * b) / (b + c)))
    fi

    echo $((x))
}
```

3 比較目錄

想法：一樣嘗試過直接parse `diff -d` 的輸出（如下圖），後來覺得應該先篩好再比較，但礙於時間限制未能順利完成



```
#echo "line: $line"
if [[ $line == diff* ]]; then

    filepath=$(echo "$line" | awk '{print $NF}' | sed 's/.*dir2\///')
    file_a=$(echo "$line" | awk '{print $((NF - 1))}')
    file_b=$(echo "$line" | awk '{print $NF}')

    x=$(calculate_changes "$file_a" "$file_b") #self-defined function

    arr+=$(echo "$filepath: changed $x%")
    echo "$filepath: changed $x%"
    #echo "file_a: $file_a"
    #echo "file_b: $file_b"

elif [[ $line == Binary* ]]; then
    filepath=$(echo "$line" | awk '{print $((NF - 1))}' | sed 's/.*dir2\///')
    arr+=$(echo "$filepath: changed 100%")
    echo "$filepath: changed 100%"

elif [[ $line == Only* ]]; then
    name=$(echo "$line" | awk '{print $3}')
    filepath=$(echo "$line" | awk '{print $NF}')

    if [[ $name == "$1:" ]]; then
        arr+=$(echo "delete $filepath")
        echo "delete $filepath"

    elif [[ $name == "$2:" ]]; then
        arr+=$(echo "create $filepath")
        echo "create $filepath"
    fi
fi
one
```

4 隱藏的檔案與 Symlink

想法：在篩選array時使用readlink來代替diff進行人工判斷

5 僅比較特定的檔案

想法：利用find指令-regex option篩選出特定檔案

參考資料：

1. sed切檔案 https://www.hy-star.com.tw/tech/linux/sed/sed.html (https://www.hy-star.com.tw/tech/linux/sed/sed.html)
2. check symlink https://koenwoortman.com/bash-script-check-if-file-is-symlink/ (https://koenwoortman.com/bash-script-check-if-file-is-symlink/)
3. https://stackoverflow.com/questions/10929453/read-a-file-line-by-line-assigning-the-value-to-a-variable (https://stackoverflow.com/questions/10929453/read-a-file-line-by-line-assigning-the-value-to-a-variable)
4. Diff command in Linux https://www.nielit.gov.in/gorakhpur/sites/default/files/Gorakhpur/alevel_unix_07_apr20_AKM.pdf (https://www.nielit.gov.in/gorakhpur/sites/default/files/Gorakhpur/alevel_unix_07_apr20_AKM.pdf)
5. 資料型態轉型 https://blog.csdn.net/abcnull/article/details/106385999 (https://blog.csdn.net/abcnull/article/details/106385999)
6. Array操作https://www.puritys.me/docs-blog/article-96-shell-script-Array.html