```bash
#!/bin/bash
#======================================================================
# SYNOPSIS
#    web_grep - find whether a web page has a certain string
#
# AUTHOR
#    Mikaela Montaos
#
# DESCRIPTION
#    service manipulation
#
# EXAMPLE
#    See below
#
# RETURN VALUE
#    Return 0 if successful
#    Return 1 otherwise.
#
# FUTURE ENHANCEMENTS
#    None
#
# BUGS
#     None
#===================================================================


#=======================================================
#
# online help
#
#=======================================================
help() {
    echo "
Description:
  service manipulation

Syntax:
  web_grep -s <string> -u <start_url> -l <number_of_levels>

Options:
    -s string
       the string to be searched
    -u start_url
       recursively search the string from this webpage and the links mentioned in
the webpage
    -l number_of_levels
       how deep is the search

Examples:
```

```
    $ web_grep -s student -u a.html -l 2
    a.html student
    a_1.html My students

    $ web_grep -s teachers -u a.html -l 3
    a_2.html Your teachers
    a_1_1.html NPU teachers
"
    exit 0
}


#========================================================================
#   main program
#========================================================================
typeset string start_url number_of_levels

if test $# -eq 0
then
   help
else
   string=
   start_url=
   number_of_levels=

   while [ "$#" -ne 0 ]
   do
      case "$1" in
         -s) shift; string="$1"; shift;;
         -u) shift; start_url="$1"; shift;;
         -l) shift; number_of_levels="$1"; shift;;

          *) echo "Error in $0: wrong option $1"; help; exit 1;;
      esac
   done
fi

##############################################
#
# Setup temporary file
#
##############################################
tmpname=${0##*/}
# if [ $(echo $tmpname | wc -c) -gt 8 ]
# then
#   typeset -L8 command=$tmpname
# else
#    command=$tmpname
```

```
# fi
# Keep the first 10 characters
command=`echo "$tmpname" | sed 's/^\(..........\).*/\1/'`
TEMPA=/tmp/${command}A$$
TEMPB=/tmp/${command}B$$
TEMPC=/tmp/${command}C$$
TEMPD=/tmp/${command}D$$
TEMPE=/tmp/${command}E$$
TEMPF=/tmp/${command}F$$
ALL_TEMP="$TEMPA $TEMPB $TEMPC $TEMPD $TEMPE $TEMPF"

trap "/bin/rm -f $ALL_TEMP; exit 1" \
        1 2 3 4 5 6 7 8 10 12 13 15 16 17 19 20 21
trap "/bin/rm -f $ALL_TEMP" 0


################################################
#
# Processing
#
################################################
# TEMPC stores all the URLs need to be processed at this level
echo "$start_url" > $TEMPC
while true
do
    if [ "$number_of_levels" = 0 ]
    then
        break
    fi

    # TEMPD is used to store the URLs to be processed at next level
    echo "" > $TEMPD

    # TEMPC stores all the URLs need to be processed at this level
    # Process one level
    cat $TEMPC | while read url
    do
        # echo current_level: $number_of_levels
        # echo $url

        lynx -dump "$url" > $TEMPA
        grep "$string" $TEMPA | while read line
        do
            echo "$url $line"
        done

        cat $TEMPA | sed -n '/^References/,$p'  | sed '1,2d' | while read num
next_url
```

```
        do
            echo $next_url
        done >> $TEMPD
    done

    # Preparing for the next level
    cp $TEMPD $TEMPC

    (( number_of_levels = number_of_levels - 1 ))
    # echo $number_of_levels
done
```

```
mva456@DESKTOP-OE7HLB3:~/cs522/inverted-index$ lynx -dump a.html
   student
   [1]a_1
   [2]a_2

References

   1. file:///home/mva456/cs522/inverted-index/a_1.html
   2. file:///home/mva456/cs522/inverted-index/a_2.html
```