

דוגמאות שהוצגו בהרצאה 12 בקורס יסודות מערכות פתוחות

1 דוגמה

שימוש ב - ברשימת שמות קבצים. דוגמאות בעמודים 84-85.

2 דוגמה

פתרון תרגיל בית 7.1.

להלן התכנית P7.1 שהוצגה בכיתה (שונה מזו שנמצאת בפתרון של התרגיל).

```
function num_word_in_dir {
    local word=$1 dir=$2 file cnt=0
    for file in $(find $dir -type f); do
        if [ $(num_word_in_file $word $file) -gt 0 ];then
            ((cnt++))
        fi
    done
    echo $cnt
}
function num_word_in_file {
    local word=$1 file=$2
    echo $(cat $file) | tr " " "\n" | egrep -c "^${word}$"
}
function all_nums_non_zero {
    local nums="$1" result x
    result=YES
    x=$(echo $nums | tr " " "\n" | egrep -c "^0$")
    if [ $x -gt 0 ]; then
        result=NO
    fi
    echo $result
}
```

```

words=""
for par in "$@"; do
    if [ $par = "-dirs" ]; then
        shift;dirs=$@;break
    fi
    words="$words $par"
    shift
done
for word in $words; do
    result="$word"
    nums=""
    for dir in $dirs; do
        nums="$nums $(num_word_in_dir $word $dir)"
    done
    if [ $(all_nums_non_zero "$nums") = YES ]; then
        echo "${result}${nums} ALL"
    else
        echo "${result}${nums} NOTALL"
    fi
done

```

```

function num_word_in_dir {
  local word=$1 dir=$2 file cnt=0
  for file in $(find $dir -type f); do
    if [ $(num_word_in_file $word $file) -gt 0 ];then
      ((cnt++))
    fi
  done
  echo $cnt
}
function num_word_in_file {
  local word=$1 file=$2
  echo $(cat $file| tr " " "\n" | egrep -c "^${word}$")
}
function all_nums_non_zero {
  local nums="$1" result x
  result=YES
  x=$(echo $nums | tr " " "\n" | egrep -c "^0$")
  if [ $x -gt 0 ]; then
    result=NO
  fi
  echo $result
}
words=""
for par in "$@"; do
  if [ $par = "-dirs" ]; then
    shift;dirs=$@;break
  fi
  words="$words $par"
  shift
done
for word in $words; do
  result="$word"
  nums=""
  for dir in $dirs; do
    nums="$nums $(num_word_in_dir $word $dir)"
  done
  if [ $(all_nums_non_zero "$nums") = YES ]; then
    echo "${result}${nums} ALL"
  else
    echo "${result}${nums} NOTALL"
  fi
done

```

3 דוגמה

פתרון תרגיל 7.2.

להלן התכנית P7.2 שהוצגה בכיתה:

```
sed -r '/^[ ]*([ ]+){3}[ ]+[*$!/d' $1 >| tmp
sed -r 's/^( [ ]*) ([ ]+)( [ ]|$)/\1\2 \2 \2\3/' tmp
```

ובפורמט טקסט:

```
sed -r '/^[ ]*([ ]+){3}[ ]+[*$!/d' $1 >| tmp
sed -r 's/^( [ ]*) ([ ]+)( [ ]|$)/\1\2 \2 \2\3/' tmp
```

4 דוגמה

פתרון שאלה 7.3. הפתרון שהוצג בכיתה זהה לפתרון שנמצא בפתרון תרגיל בית 7 שבאתר הקורס.

5 דוגמה

פתרון שאלה 7.4. (שונה מזה שמוצג בפתרון התרגיל). בפתרון שהוצג בכיתה היה חסר מימוש של מספר פונקציות (וגם היו כמה שגיאות שהתגלו בזמן הרצת התכנית).

להלן הפתרון שהוצג בכיתה בתוספת הפונקציות החסרות ולאחר תיקון שגיאות שהתגלו בזמן ההרצה.

```
function num_word_in_files {
    local word=$1 files="$(echo $@|cut -d" " -f2-)"
    echo $(cat $files)| tr " " "\n" | egrep -c "^${word}$"
}
function get_rows {
    local file=$1 start=$2 end=$3 n
    n=$(wc -l<$file)
    if [ $start -gt $n ]; then
        return
    fi
    if [ $end -gt $n ]; then
        end=$n
    fi
    head -$end $file | tail -${(end-start)+1}
}
}
```

```

function get_cols {
  local file=$1 start=$2 end=$3 n
  while read line; do
    n=$(echo $line | wc -w)
    if [ $start -gt $n ]; then
      continue
    fi
    echo $line | cut -d" " -f${start}-${end}
  done <$file
}

function get_rows_cols {
  local file=$1 row1=$2 row2=$3 col1=$4 col2=$5
  get_rows $file $row1 $row2 >| tmp
  get_cols tmp $col1 $col2
}

function intersection {
  local files="$@" num_files=$#
  for file in $files; do
    echo $(cat $file) | tr " " "\n" | sort -u
  done >|tmp
  sort tmp | uniq -c >| tmp1
  while read line; do
    echo $line
  done <tmp1 >| tmp2
  echo $(egrep "^${num_files} " tmp2 | cut -d" " -f2)
}

row1=$1;row2=$2;col1=$3;col2=$4
files=$(echo $@ | cut -d" " -f5-)
files1=""
for file in $files; do
  get_rows_cols $file $row1 $row2 $col1 $col2 >| ${file}.1
  files1="$files1 ${file}.1"
done
words=$(intersection $files1)
for word in $words; do
  echo $word $(num_word_in_files $word $files)
done>|tmp
sort -k 2nr,2 -k 1r,1 tmp

```

```

function num_word_in_files {
    local word=$1 files="$(echo $@|cut -d" " -f2-)"
    echo $(cat $files| tr " " "\n" | egrep -c "^${word}$")
}
function get_rows {
    local file=$1 start=$2 end=$3 n
    n=$(wc -l<$file)
    if [ $start -gt $n ]; then
        return
    fi
    if [ $end -gt $n ]; then
        end=$n
    fi
    head -$end $file | tail -${(end-start)+1}
}
function get_cols {
    local file=$1 start=$2 end=$3 n
    while read line; do
        n=$(echo $line | wc -w)
        if [ $start -gt $n ]; then
            continue
        fi
        echo $line | cut -d" " -f${start}-${end}
    done <$file
}
function get_rows_cols {
    local file=$1 row1=$2 row2=$3 col1=$4 col2=$5
    get_rows $file $row1 $row2 >| tmp
    get_cols tmp $col1 $col2
}
function intersection {
    local files="$@" num_files=$#
    for file in $files; do
        echo $(cat $file) | tr " " "\n" | sort -u
    done >|tmp
    sort tmp | uniq -c >| tmp1
    while read line; do
        echo $line
    done <tmp1 >| tmp2
    echo $(egrep "^${num_files}" tmp2 | cut -d" " -f2)
}
row1=$1;row2=$2;col1=$3;col2=$4
files="$(echo $@ | cut -d" " -f5-)"
files1=""
for file in $files; do
    get_rows_cols $file $row1 $row2 $col1 $col2 >| ${file}.1
    files1="${files1} ${file}.1"
done
words="$(intersection $files1)"
for word in $words; do
    echo $word $(num_word_in_files $word $files)
done>|tmp
sort -k 2nr,2 -k 1r,1 tmp

```