

Име: _____ ФН: _____ Спец.: __ Курс: __ Гр.: __

Задача	1 (30)	2 (30)	3 (30)	4 (30)	Общо (120)
получени точки					

Забележка: За всички задачи, в имената на файловете и директориите няма специални символи. Във файловата система може да съществуват директории, до които нямате достъп като обикновен потребител.

Задача 1. Напишете скрипт, който извежда името на потребителския акаунт, в чиято home директория има най-скоро променен обикновен файл и кой е този файл. Напишете скрипта с подходящите проверки, така че да бъде валиден администраторски инструмент.

Задача 2. Напишете скрипт, който приема три задължителни позиционни аргумента:

- име на файл
- *низ1*
- *низ2*

Файлът е текстови, и съдържа редове във формат:

ключ=стойност

където *стойност* може да бъде:

- празен низ, т.е. редът е *ключ=*
- низ, състоящ се от един или повече термове, разделени с интервали, т.е., редът е *ключ=t₁ t₂ t₃*

Някъде във файла:

- се съдържа един ред с *ключ* първия подаден низ (*низ1*);
- и може да се съдържа един ред с *ключ* втория подаден низ (*низ2*).

Скриптът трябва да *променя реда с ключ низ2* така, че обединението на термовете на редовете с ключове *низ1* и *низ2* да включва всеки терм еднократно.

Примерен входен файл:

```
$ cat z1.txt
FOO=73
BAR=42
BAZ=
ENABLED_OPTIONS=a b c d
ENABLED_OPTIONS_EXTRA=c e f
```

Примерно извикване:

```
$ ./a.sh z1.txt ENABLED_OPTIONS ENABLED_OPTIONS_EXTRA
```

Изходен файл:

```
$ cat z1.txt
FOO=73
BAR=42
BAZ=
ENABLED_OPTIONS=a b c d
ENABLED_OPTIONS_EXTRA=e f
```

Задача 3. Нека съществува програма за моментна комуникация (Instant messaging), която записва логове на разговорите в следния формат:

- има определена директория за логове (*LOGDIR*)
- в нея има директорийна структура от следния вид:
LOGDIR/протокол/акаунт/приятел/
като на всяко ниво може да има няколко екземпляра от съответния вид, т.е. няколко директории *протокол*, във всяка от тях може да има няколко директории *акаунт*, и във всяка от тях – няколко директории *приятел*
- във всяка от директориите *приятел* може да има файлове с имена от вида *yyyy-mm-dd-hh-mm-ss.txt* – година-месец-ден и т.н., спрямо това кога е започнал даден разговор
- всеки такъв файл представлява лог на даден разговор със съответния приятел, като всяка разменена реплика между вас е на отделен ред
- даден идентификатор *приятел* може да се среща няколко пъти в структурата (напр. през различни ваши акаунти сте водили разговори със същия приятел)

Напишете скрипт, който приема задължителен позиционен аргумент - име на лог директория (*LOGDIR*). Скриптът трябва да извежда десетимата приятели, с които имате най-много редове комуникация глобално (без значение протокол и акаунт), и колко реда имате с всеки от тях. Опишете в коментар как работи алгоритъмът ви.

Задача 4. Дадена е програма за ОС Linux, написана на езика C:

```
#include <unistd.h>
#include <stdio.h>
int main(void)
{
    int p1, p2;
    p1=fork();
    p2=fork();
    printf("Hello world!\n");
}
```

- Колко пъти ще се отпечата текста "Hello world!" при изпълнението на програмата? Обосновете отговора си.
- Как работи системното извикване `fork()`?
- Нарисувайте кореновото дърво с върхове процесите, които ще се стартират в резултат от изпълнението на програмата и ребра двойките родител-наследник.

Примерни решения

Задача 1.

```
#!/bin/bash

if [ $(id -u) -ne 0 ]; then
    echo "should be started as root"
    exit 1
fi

while read US HD; do
    if [ -d "${HD}" ]; then
        find "${HD}" -type f -printf "${US} %T@ %p\n"
    fi
done < <(cat /etc/passwd | cut -d ':' -f1,6 | tr ':' '\t') \
    | sort -k2 -rn | head -n1 | cut -d ' ' -f 1,3
```

Задача 2.

```
#!/bin/bash
[ $# -eq 3 ] || exit 1

S1=$(egrep "^$2=" $1 | cut -d '=' -f 2)

if [ $(egrep -c "^$3=" $1) -eq 0 ]; then
    echo "$3=" >> $1
    exit 0
fi

S2=$(egrep "^$3=" $1 | cut -d '=' -f 2)
NEWS2=$(comm -13 <(echo "$S1" | tr -s ' ' | tr ' ' '\n' | sort) \
    <(echo $S2 | tr -s ' ' | tr ' ' '\n' | sort) | xargs)
sed -i -e "s/^$3=$S2/$3=$NEWS2/" $1
```

Задача 3.

```
#!/bin/bash

[ $# -ne 1 ] && echo "Usage: $0 datadir" 1>&2 && exit 1

data="$1"
fpattern='????-??-??-??-??-??.txt'

[ -d "${data}" ] || exit 2

declare -A arr

for abuddy in $(find -- "${data}" -mindepth 3 -maxdepth 3 -type d); do
    buddy="$(basename -- "${abuddy}")"
    lines=$(find -- ${abuddy} -type f -name ${fpattern} | xargs cat | wc -l)
    arr[${buddy}]=$(( ${arr[${buddy}]:-0} + ${lines} ))
done

for buddy in "${!arr[@]}"; do
    echo ${arr[${buddy}]} ${buddy}
done | sort -rn | head
```



```
#!/bin/bash
```

```
[ $# -ne 1 ] && echo "Usage: $0 datadir" 1>&2 && exit 1
```

```
data="$(dirname -- ${1})/${basename ${1}}"
```

```
fpat='????-??-??-??-??-???.txt'
```

```
[ -d "${data}" ] || exit 2
```

```
for buddy in $(find -- "${data}" -mindepth 3 -maxdepth 3 -type d \  
  -exec basename {} \; | sort | uniq ); do
```

```
  echo "$(cat ${data}/*//${buddy}/${fpat} | wc -l) ${buddy}"  
done | sort -rn | head
```