BASH练习题

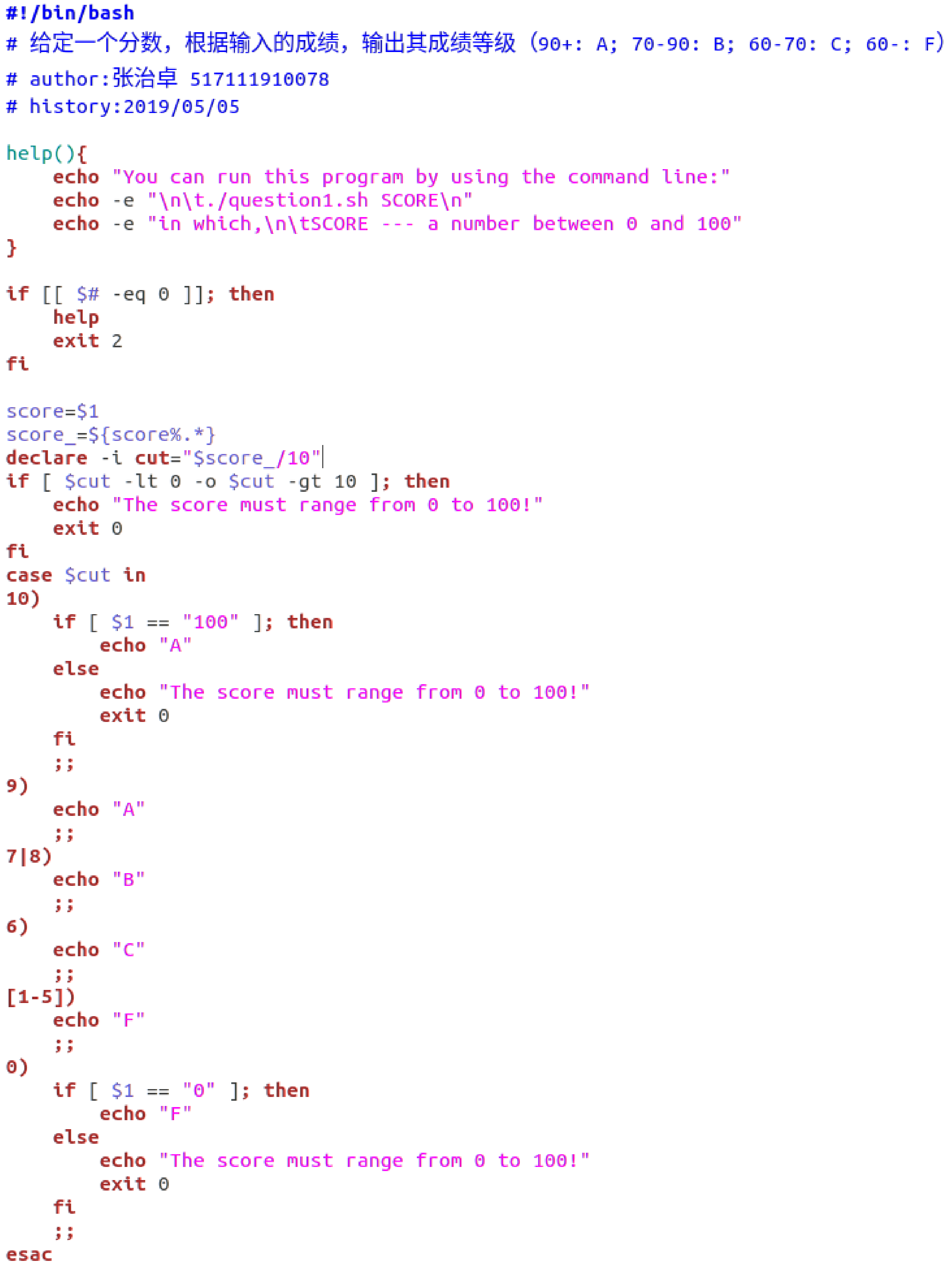
注意对于输入的命令行参数进行检查（你可以设置自己的错误号码表）

---

1. 给定一个分数，根据输入的成绩，输出其成绩等级（90+: A; 70-90: B; 60-70: C; 60-: F）

```
user@host ~ $ ./scorelev.sh 98
A
user@host ~ $ ./scorelev.sh 45
F
```

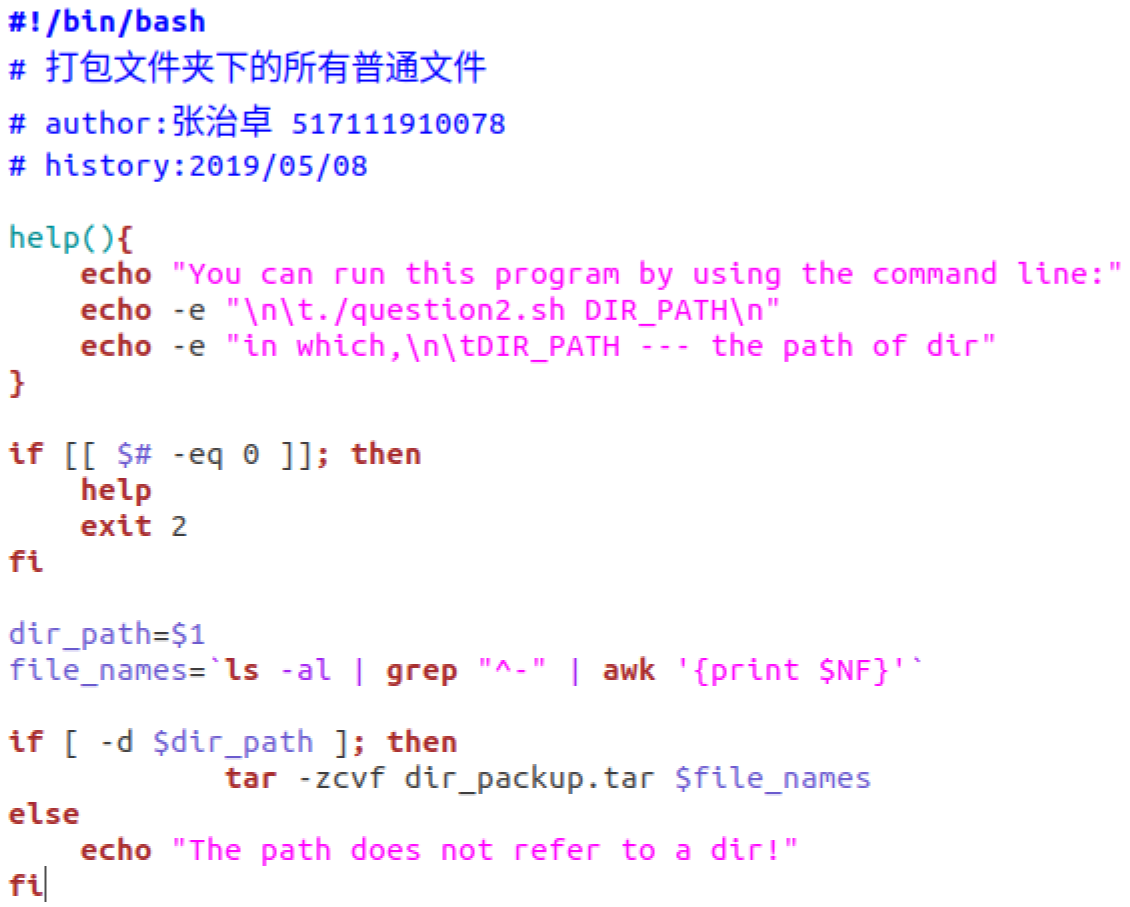
（提示：使用case，还有必须判断该输入成绩的范围是否合理，当然if也可以）



```
#!/bin/bash
# 给定一个分数，根据输入的成绩，输出其成绩等级（90+: A; 70-90: B; 60-70: C; 60-: F）
# author:张治卓 517111910078
# history:2019/05/05

help(){
    echo "You can run this program by using the command line:"
    echo -e "\n\t./question1.sh SCORE\n"
    echo -e "in which,\n\tSCORE --- a number between 0 and 100"
}

if [[ $# -eq 0 ]]; then
    help
    exit 2
fi

score=$1
score_=${score%.*}
declare -i cut="$score_/10"
if [ $cut -lt 0 -o $cut -gt 10 ]; then
    echo "The score must range from 0 to 100!"
    exit 0
fi
case $cut in
10)
    if [ $1 == "100" ]; then
        echo "A"
    else
        echo "The score must range from 0 to 100!"
        exit 0
    fi
    ;;
9)
    echo "A"
    ;;
7|8)
    echo "B"
    ;;
6)
    echo "C"
    ;;
[1-5])
    echo "F"
    ;;
0)
    if [ $1 == "0" ]; then
        echo "F"
    else
        echo "The score must range from 0 to 100!"
        exit 0
    fi
    ;;
esac
```

generated by haroopad

2. 写一个bash脚本，根据输入的目录名，对目录下的**普通文件**进行备份、打包



```
#!/bin/bash
# 打包文件夹下的所有普通文件
# author:张治卓 517111910078
# history:2019/05/08

help(){
    echo "You can run this program by using the command line:"
    echo -e "\n\t./question2.sh DIR_PATH\n"
    echo -e "in which,\n\tDIR_PATH --- the path of dir"
}

if [[ $# -eq 0 ]]; then
    help
    exit 2
fi

dir_path=$1
file_names=`ls -al | grep "^-" | awk '{print $NF}'`

if [ -d $dir_path ]; then
            tar -zcvf dir_packup.tar $file_names
else
    echo "The path does not refer to a dir!"
fi
```

3. 课上我们介绍过如何去处理一个字符串，获取其中特定的子串，这里需要大家写一个函数，解析输入的格式为”12/28/2012”这种日期的字符串，分别获取年份、月份、日，并将其处理成另外一种写法进行输出。

```
###### 利用数组 ######
# 日期变量赋值
day="12/28/2012"
# 设定分隔符
IFS="/"
# 声明数组变量
declare -a dd
dd=($day)
# 用分隔符转换为数组
declare -p dd
# 输出年
echo "Year: ${dd[2]}"
echo "Month: ${dd[1]}"
echo "Day: ${dd[0]}"
```

```
#!/bin/bash
# 输出年月日
# author:张治卓 517111910078
# history:2019/05/08

help(){
    echo "You can run this program by using the command line:"
    echo -e "\n\t./question3.sh DATE\n"
    echo -e "in which,\n\tDATE --- date in pattern <m/d/y>"
}

if [[ $# -eq 0 ]]; then
    help
    exit 2
fi

day=$1
IFS="/"
declare -a dd
dd=($day)

if [ ${dd[2]} -lt 0 ]; then
    echo "Year is out of range!"
    exit 0
fi
if [ ${dd[0]} -lt 1 -o ${dd[0]} -gt 12 ]; then
    echo "Month is out of range!"
    exit 0
fi
case ${dd[0]} in
1|3|5|7|8|10|12)
    if [ ${dd[1]} -lt 1 -o ${dd[1]} -gt 31 ]; then
        echo "Day is out of range!"
        exit 0
    fi
    ;;
4|6|9|11)
    if [ ${dd[1]} -lt 1 -o ${dd[1]} -gt 30 ]; then
        echo "Day is out of range!"
        exit 0
    fi
    ;;
2)
    a=`echo "${dd[2]}%4" | bc`
    b=`echo "${dd[2]}%100" | bc`
    if [ $a -eq 0 -a $b -ne 0 ]; then
        if [ ${dd[1]} -lt 1 -o ${dd[1]} -gt 29 ]; then
            echo "Day is out of range!"
            exit 0
        fi
    else
        if [ ${dd[1]} -lt 1 -o ${dd[1]} -gt 28 ]; then
            echo "Day is out of range!"
            exit 0
        fi
    fi
    ;;
    ;;
esac

echo "Year: ${dd[2]}"
echo "Month: ${dd[0]}"
echo "Day: ${dd[1]}"
```

4. 针对 `/etc/passwd` 文件，写出所有默认登录shell为 `/bin/bash` 的用户，并输出下面的语句。

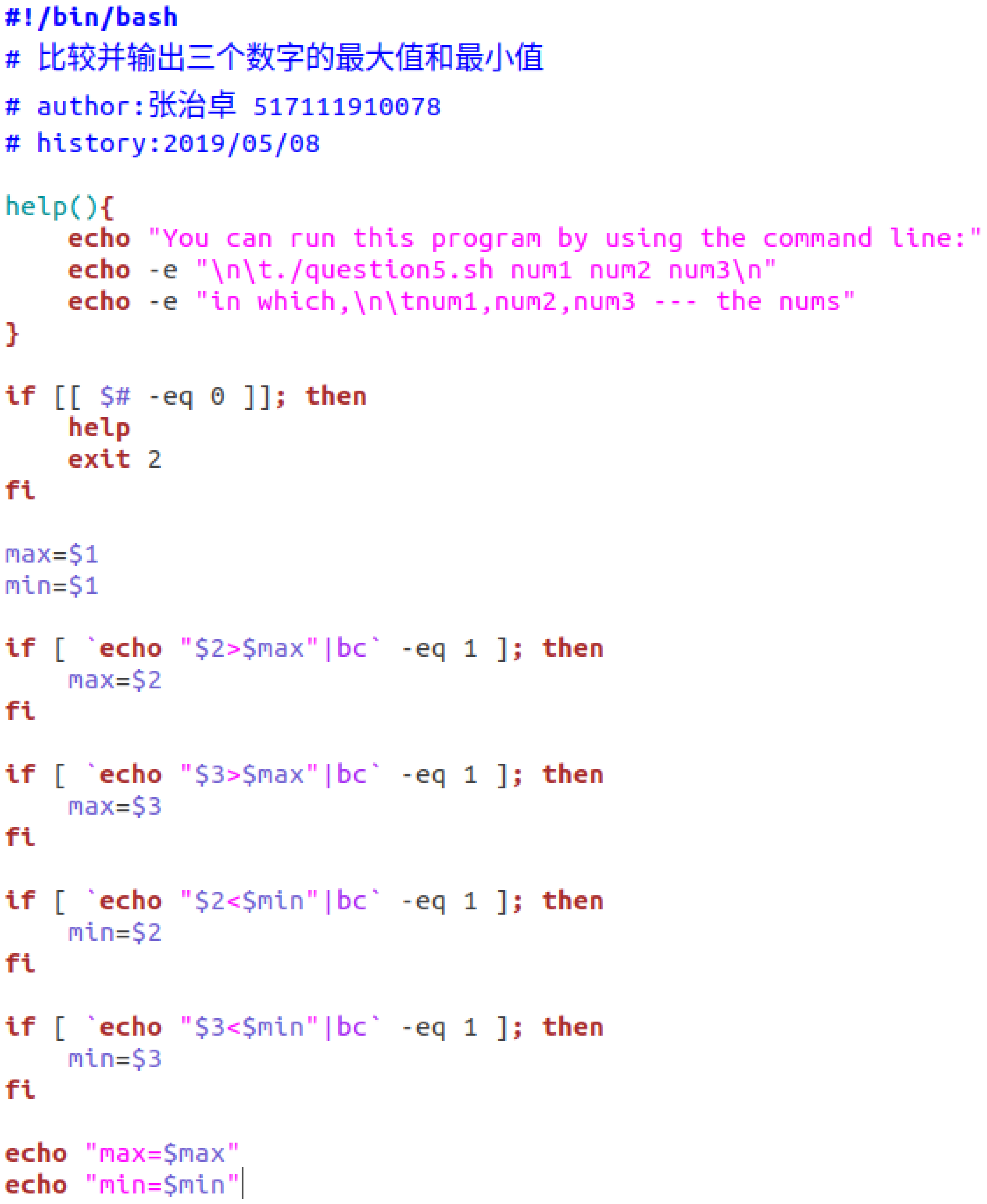
```
root is bash user
bio is bash user
bioinfo is bash user
```

(提示：使用while/if/read重定向实现)

```
#!/bin/bash
# 输出登录shell为bash的用户
# author:张治卓 517111910078
# history:2019/05/08

IFS=":"

while read f1 f2 f3 f4 f5 f6 f7
do
    if [ $f7 == "/bin/bash" ]; then
        echo "$f1 is bash user"
    fi
done < /etc/passwd
```

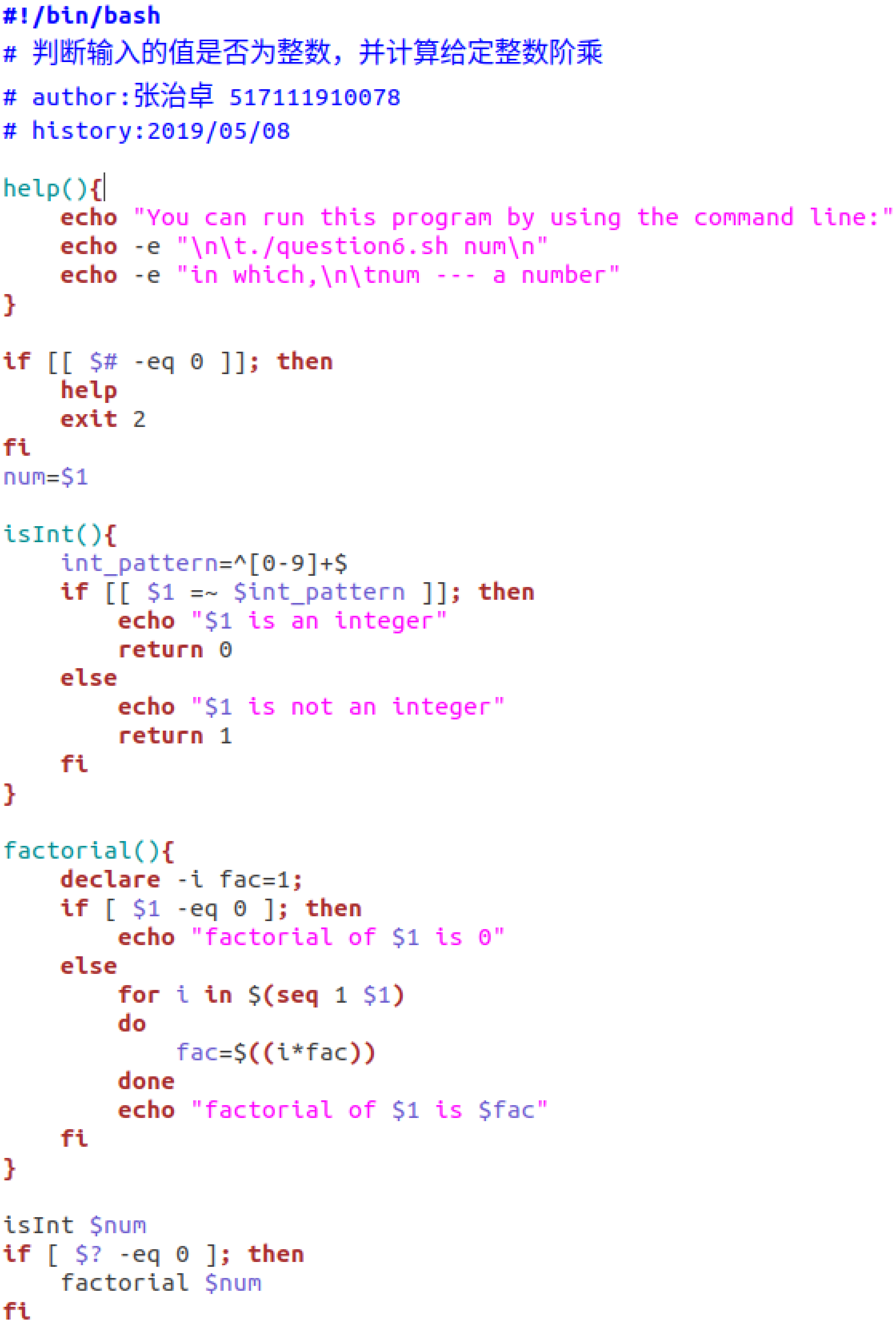
5. 从键盘输入三个数a、b、c，并输出其中的最大值和最小值，可以在一行上输入，也可以分别获取。

（提示：注意read的用法，或者直接在命令行输入获取）



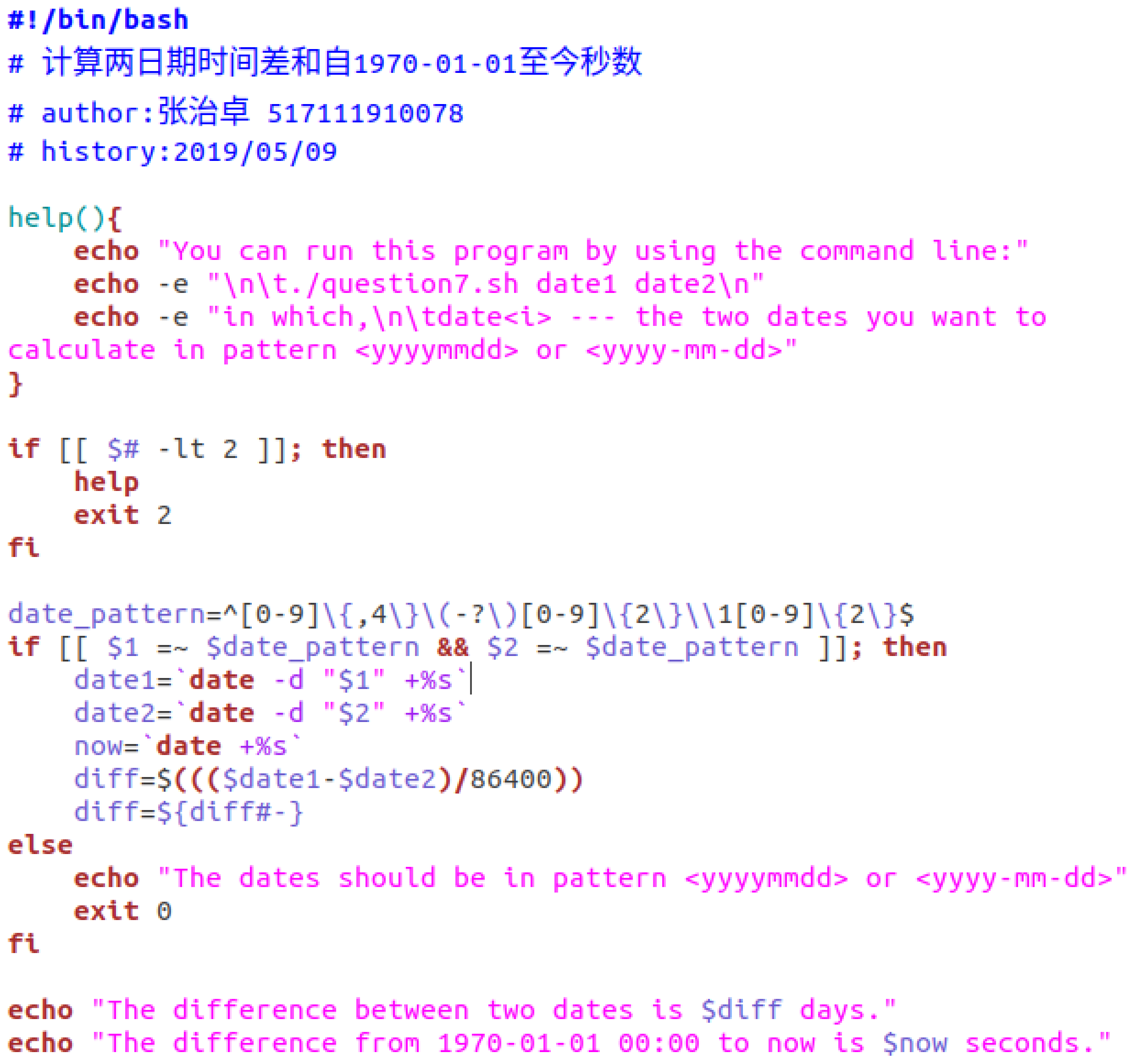
```
#!/bin/bash
# 比较并输出三个数字的最大值和最小值
# author:张治卓 517111910078
# history:2019/05/08

help(){
    echo "You can run this program by using the command line:"
    echo -e "\n\t./question5.sh num1 num2 num3\n"
    echo -e "in which,\n\tnum1,num2,num3 --- the nums"
}

if [[ $# -eq 0 ]]; then
    help
    exit 2
fi

max=$1
min=$1

if [ `echo "$2>$max"|bc` -eq 1 ]; then
    max=$2
fi

if [ `echo "$3>$max"|bc` -eq 1 ]; then
    max=$3
fi

if [ `echo "$2<$min"|bc` -eq 1 ]; then
    min=$2
fi

if [ `echo "$3<$min"|bc` -eq 1 ]; then
    min=$3
fi

echo "max=$max"
echo "min=$min"
```

6. 写一个函数，判断输入的值是否为整数，并计算给定整数阶乘（factorial）



```
#!/bin/bash
# 判断输入的值是否为整数，并计算给定整数阶乘
# author:张治卓 517111910078
# history:2019/05/08

help(){
    echo "You can run this program by using the command line:"
    echo -e "\n\t./question6.sh num\n"
    echo -e "in which,\n\tnum --- a number"
}

if [[ $# -eq 0 ]]; then
    help
    exit 2
fi
num=$1

isInt(){
    int_pattern=^[0-9]+$
    if [[ $1 =~ $int_pattern ]]; then
        echo "$1 is an integer"
        return 0
    else
        echo "$1 is not an integer"
        return 1
    fi
}

factorial(){
    declare -i fac=1;
    if [ $1 -eq 0 ]; then
        echo "factorial of $1 is 0"
    else
        for i in $(seq 1 $1)
        do
            fac=$((i*fac))
        done
        echo "factorial of $1 is $fac"
    fi
}

isInt $num
if [ $? -eq 0 ]; then
    factorial $num
fi
```

7. 计算1970年1月1日0时0分至当前时间的长度，以秒为单位。计算任意给定两时间之间的天数。（提示：date命令获取当前时间）



```
#!/bin/bash
# 计算两日期时间差和自1970-01-01至今秒数
# author:张治卓 517111910078
# history:2019/05/09

help(){
    echo "You can run this program by using the command line:"
    echo -e "\n\t./question7.sh date1 date2\n"
    echo -e "in which,\n\tdate<i> --- the two dates you want to
calculate in pattern <yyyymmdd> or <yyyy-mm-dd>"
}

if [[ $# -lt 2 ]]; then
    help
    exit 2
fi

date_pattern=^[0-9]\{,4\}\(-?\)[0-9]\{2\}\\1[0-9]\{2\}$
if [[ $1 =~ $date_pattern && $2 =~ $date_pattern ]]; then
    date1=`date -d "$1" +%s`
    date2=`date -d "$2" +%s`
    now=`date +%s`
    diff=$((($date1-$date2)/86400))
    diff=${diff#-}
else
    echo "The dates should be in pattern <yyyymmdd> or <yyyy-mm-dd>"
    exit 0
fi

echo "The difference between two dates is $diff days."
echo "The difference from 1970-01-01 00:00 to now is $now seconds."
```

8. 有两个目录，其中一个目录是另外一个的拷贝（备份），根据其所有对应文件的差别对其制作出补丁（patch），全部输出到一个补丁文件 `%DATE.patch` 中，其中%DATE为当前的日期，比如

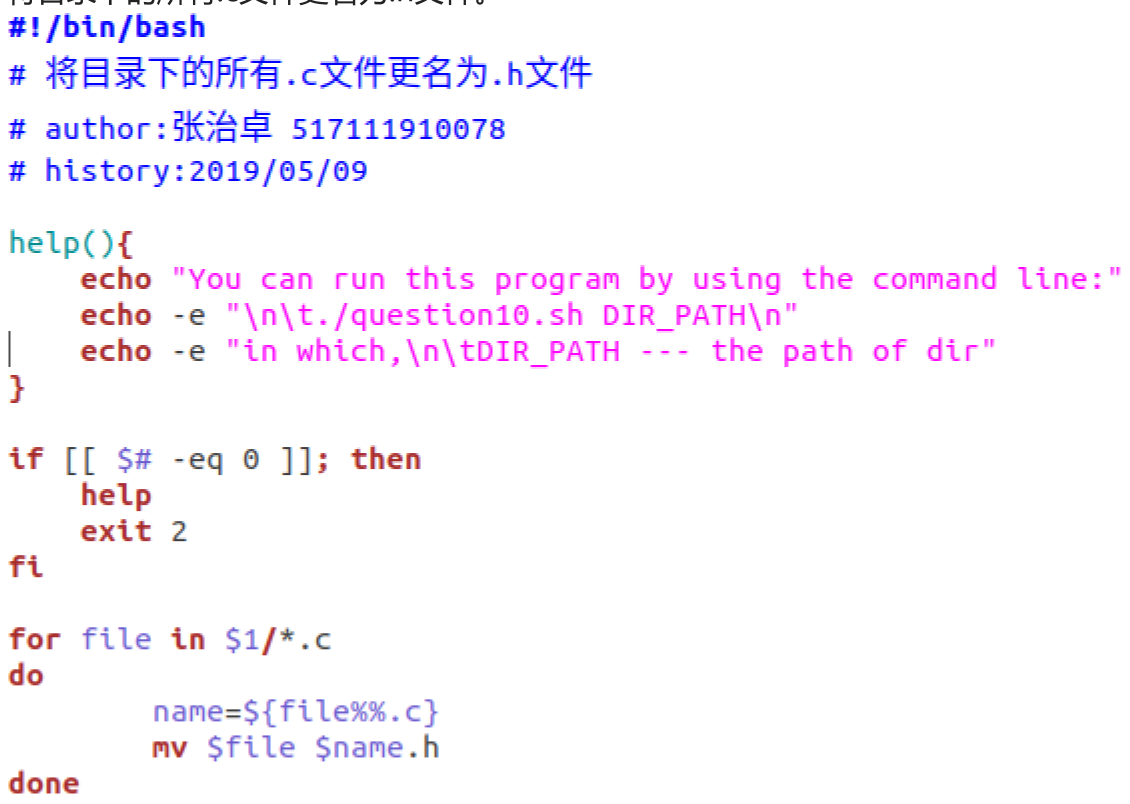
20130327.patch

```bash
#!/bin/bash
# 根据其所有对应文件的差别对其制作出补丁（patch）
# author:张治卓 517111910078
# history:2019/05/09

help(){
    echo "You can run this program by using the command line:"
    echo -e "\n\t./question8.sh dir dir_copy\n"
    echo -e "in which,\n\tdir --- the original directory\n\tdir_copy --- the copy of the former directory"
}

if [[ $# -lt 2 ]]; then
    help
    exit 2
fi

dir=$1
dir_copy=$2
date=$(date +%Y%m%d)
echo "Generally difference:" > ${date}.patch
diff -uq $dir $dir_copy >> ${date}.patch
echo -e "\n\n\nSpecific difference:" >> ${date}.patch
diff -ur $dir $dir_copy >> ${date}.patch
```

9. 假设现在有一个文本文件，能不能用一个bash脚本，将其中的所有大写字母转换为小写字母，并将其写回该文件？

```bash
#!/bin/bash
# 将所有大写字母转换为小写字母，并将其写回文件
# author:张治卓 517111910078
# history:2019/05/09

help(){
    echo "You can run this program by using the command line:"
    echo -e "\n\t./question9.sh FILE_PATH\n"
    echo -e "in which,\n\tFILE_PATH --- the path of file"
}

if [[ $# -eq 0 ]]; then
    help
    exit 2
fi

if [[ -f $1 ]]; then
    cat $1 | tr -t 'a-z' 'A-Z' > .bash_temp
    cat .bash_temp > $1
    rm .bash_temp
else
    echo "File doesn't exist!"
fi
```

10. 将目录下的所有.c文件更名为.h文件。



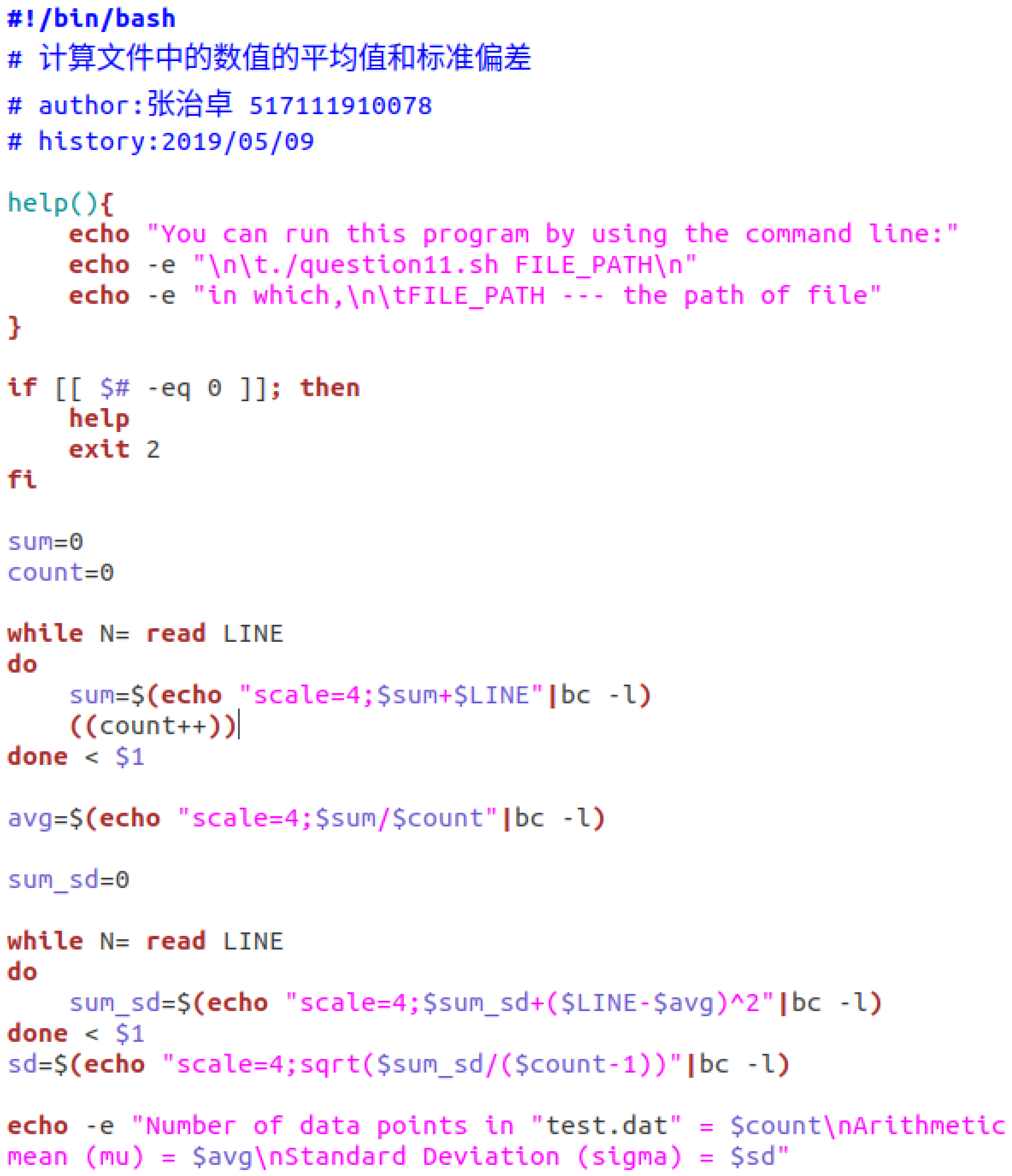
```
#!/bin/bash
# 将目录下的所有.c文件更名为.h文件
# author:张治卓 517111910078
# history:2019/05/09

help(){
    echo "You can run this program by using the command line:"
    echo -e "\n\t./question10.sh DIR_PATH\n"
    echo -e "in which,\n\tDIR_PATH --- the path of dir"
}

if [[ $# -eq 0 ]]; then
    help
    exit 2
fi

for file in $1/*.c
do
        name=${file%%.c}
        mv $file $name.h
done
```

11. 写一个函数sd.sh，计算下面文件中的数值的平均值和标准偏差。

```
# test.dat
12.33
11.67
12.16
13.01
12.56
```

下面是运行示例：

```
user@host ~$ ./sd.sh test.dat
Number of data points in "test.dat" = 5
Arithmetic mean (mu) = 12.1234
Standard Deviation (sigma) = .3456
```



```
#!/bin/bash
# 计算文件中的数值的平均值和标准偏差
# author:张治卓 517111910078
# history:2019/05/09

help(){
    echo "You can run this program by using the command line:"
    echo -e "\n\t./question11.sh FILE_PATH\n"
    echo -e "in which,\n\tFILE_PATH --- the path of file"
}

if [[ $# -eq 0 ]]; then
    help
    exit 2
fi

sum=0
count=0

while N= read LINE
do
    sum=$(echo "scale=4;$sum+$LINE"|bc -l)
    ((count++))
done < $1

avg=$(echo "scale=4;$sum/$count"|bc -l)

sum_sd=0

while N= read LINE
do
    sum_sd=$(echo "scale=4;$sum_sd+($LINE-$avg)^2"|bc -l)
done < $1
sd=$(echo "scale=4;sqrt($sum_sd/($count-1))"|bc -l)

echo -e "Number of data points in "test.dat" = $count\nArithmetic
mean (mu) = $avg\nStandard Deviation (sigma) = $sd"
```

12. 密码检查

写一个脚本，检查你输入的密码是“强（strong）” 、 “中等强（medium）” 还是“弱（weak）” 口令，根据下面的标准：

- 长度至少为8个字符；
- 至少包含一个大写字母，一个小写字母和一个数字
- 至少包含一个特殊字符：@, #, $, %, &, *, +, -, =

三项满足的是强口令，满足两项的为中等强度口令，否则为弱口令，分别写出对应实例进行验

证。

```
#!/bin/bash
# 检查输入的密码是“强（strong）”、“中等强（medium）”还是“弱（weak）”口令
# author:张治卓 517111910078
# history:2019/05/09

help(){
    echo "You can run this program by using the command line:"
    echo -e "\n\t./question12.sh PASSWD\n"
    echo -e "in which,\n\tPASSWD --- the password you want"
}

if [[ $# -eq 0 ]]; then
    help
    exit 2
fi

num=0
low=0
upp=0
spe=0
count=$(echo $1 | wc -m)
((count--))
spe_pattern=[=@#$%\&*+-]

if [[ $1 =~ [0-9] ]]; then num=1; fi
if [[ $1 =~ [a-z] ]]; then low=1; fi
if [[ $1 =~ [A-Z] ]]; then upp=1; fi
if [[ $1 =~ $spe_pattern ]]; then spe=1; fi

r1=0
r2=0
r3=0

if [ $count -ge 8 ]; then r1=1; fi
if [ $num -eq 1 -a $low -eq 1 -a $upp -eq 1 ]; then r2=1; fi
if [ $spe -eq 1 ]; then r3=1; fi

declare -i r=$r1+$r2+$r3

case $r in
0|1)
    echo "$1 is a weak password"
    ;;
2)
    echo "$1 is a medium password"
    ;;
3)
    echo "$1 is a strong password"
    ;;
esac
```