

Checkfile.sh

#!/bin/bash

#!/bin/bash

check_file.sh

Usage: ./check_file.sh filename.txt

if [\$# -ne 1]; then

echo "Usage: \$0 <filename>"

exit 1

fi

file = "\$1"

if [-e "\$file"]; then

echo "File exists: \$file"

echo "---- contents ----"

cat -- "\$file"

else

echo "File '\$file' does not exist."

read -p "Create it now? (y/n): " ans

case "\$ans" in

[Yy]*) touch "\$file"; echo "Created \$file"

"You can edit it using your favorite editor."

*) echo "Not creating file.";;

esac

fi

PAGE NO. _____
DATE: / /

```
#!/bin/bash
# check_file.sh
# Usage: ./check_file.sh <"filename.txt"
if [ $# -ne 1 ]; then
    echo "Usage: $0 <filename>"
    exit 1
fi
file="$1"
if [ -e "$file" ]; then
    echo "File does not exists: $file"
    cat -- "$file"
else
    echo "File does not exist."
    read -p "Create it now? (y/n); " ans
    case $ans in
        [yY])
            touch "$file"; echo "Created $file"; echo
            "You can create using file editor";;
        *)
            echo "Not creating file";;
    esac
fi
```

```
#!/bin/bash  
# check-file.sh  
# Usage: ./check-file.sh filename.txt
```

```
if [ $# -ne 1 ]; then  
    echo "Usage: $0 <filename>"  
    exit 1  
fi
```

```
file = "$1"  
if [-e "$file"]; then  
    echo "file exists: $file"  
    echo "--- contents ---"  
    cat -- "$file"  
else
```

```
    echo "file '$file' does not exist"  
    read -p "Create it now? (y/n) :" ans  
    case "$ans" in
```

```
[y/Y]*) touch "$file"; echo "Created $file";  
    "You can edit it using your fav. editor";;  
    echo "Not creating file";;
```

```
esac
```

```
fi.
```

```
echo "file '$file' doesn't exist"  
read -p "Create it now? (y/n) :" ans  
case "$ans" in
```

```
[y/Y]*) touch "$file"; echo "Created $file"; echo "You can  
    using your fav. editor";;;
```

~~edit your fav~~

```
echo "Not creating file";
```

```
esac.
```

```
#!/bin/bash
# check_file.sh
# Usage: ./check_file.sh "filename.txt"
if [ $# -ne 1 ]; then
    echo "Usage: $0 <filename>""
    exit 1
fi
file= "$1"
if [ -e "$file" ]; then
    echo "file exists $file"
    echo "-- content --"
    cat -- "$file"
else
    echo "file '$file' does not exist"
    read -p "Create it now? (y/n)" ans
    case "$ans" in
        [yY])
            touch "$file"; echo "Created $file"; echo "You can edit"
            using your favorite editor"; ;;
        [nN])
            echo "Not creating file"; ;;
    esac
fi
```

```
#!/bin/bash
# check-file.sh
# Usage: ./check-file.sh "filename.txt"
if [ $# -ne 1 ]; then
    echo "Usage: $0 <filename>"
    exit 1
fi
file="$1"
if [-e "$file"]; then
    echo "file exist $file"
    echo "-- content --"
    cat -- "$file"
else
```

#!/bin/bash
count-1wc.sh
Usage: ./count_1wc.sh filename.txt

if [\$# -ne 1]; then
echo "Usage: \$0 <filename>"
exit 1

fi
if [! -f "\$1"]; then
echo "File not found!"
exit 1

fi
lines=\$ (wc -l < "\$1")
words=\$ (wc -w < "\$1")
chars=\$ (wc -m < "\$1")
echo "Lines: \$lines"
echo "Words: \$words"
echo "Chars: \$chars".

#!/bin/bash
count_1wc.sh
Usage: ./count_1wc.sh filename.txt

if [\$# -ne 1]; then
echo "Usage: \$0 <filename>"
exit 1

fi
if [! -f "\$1"]; then
echo "File not found!"
exit 1

fi
lines=\$ (wc -l < "\$1")
words=\$ (wc -w < "\$1")
chars=\$ (wc -m < "\$1")

} {
echo "Lines: \$lines"
echo "Words: \$words"
echo "Chars: \$chars"

#!/bin/bash
 # count-lws.sh
 # Usage: ./count-lws.sh filename.txt

 if [\$# ne 1]; then
 echo "Usage: \$0 <filename>"
 exit 1
 fi
 if [! -f "\$1"]; then
 echo "file not found"
 exit 1
 fi
 line=\$(wc -l < "\$1")
 words=\$(wc -w < "\$1")
 char=\$(wc -m < "\$1")
 echo "line: \$line"
 echo "word: \$words"
 echo "character: \$char"

 #!/bin/bash
 # Count-lws.sh
 # Usage: ./count-lws.sh filename.txt
 if [\$# ne 1]; then
 echo "Usage: \$0 <filename>"
 exit 1
 fi
 if [! -f "\$1"]; then
 echo "file not found"
 exit 1
 fi
 line=\$(wc -l < "\$1")
 word=\$(wc -w < "\$1")
 char=\$(wc -m < "\$1")
 echo "line: \$line"
 echo "word: \$word"
 echo "character: \$char"

```
#!/bin/bash
# Usage: ./one_to_ten.sh
a=(1 2 3 4 5 6 7)
for i in "${a[@]}"; do
    echo "$i"
done
```

```
#!/bin/bash
# Usage: ./one_to_ten.sh
a=(1 2 3 4 5 6 7)
for i in "${a[@]}"; do
    echo "$i"
done
```

```
#!/bin/bash
# Usage: ./one_to_ten.sh
a=(1 2 3 4 5 6 7)
for i in "${a[@]}"; do
    echo "$i"
done
```

```
#!/bin/bash
# Usage: ./one_to_ten.sh
a=(1 2 3 4 5 6 7)
for i in "${a[@]}"; do
    echo "$i"
done
```

```
#!/bin/bash
# Usage: ./one_to_ten.sh
a=(1 2 3 4 5 6 7)
for i in "${a[@]}"; do
    echo "$i"
done
```

```

for ((i=2, j=n; i++)) ; do
    res=$((res * i))
done.
echo "$res"
if [[ $# -gt 1 ]]; then
    echo "Usage: $0 <non-negative-integer> [another...]"
    exit 1
fi
for arg in "$@"; do
    if ! [[ $arg =~ ^[0-9]+[k] ]]; then
        echo "$arg: nota non-negative integer, skipping."
        continue
    fi
    echo "$arg! = $(fact "$arg"))"
done.

```

```
#!/bin/bash
# factorial.sh
# Usage: ./factorial.sh n
fact() {
    n=$1
    if [[ $n -le 1 ]]; then
        echo 1
        return
    fi
    res=1
    for ((i=2; i<=n; i++)); do
        res=$((res * i))
    done
    echo "$res"
    if [[ $# -ne 1 ]]; then
        echo "Usage: $0 <non-negative integer> [another]"
        exit 1
    fi
    for arg in "$@"; do
        if ! [[ $arg =~ ^[0-9]+[ ]? ]]; then
            echo "$arg: not a -ve integer skipping"
            continue
        fi
        echo "$arg! = $(fact "$arg")"
    done
}
```

```
##!/bin/bash
## factorial with
## Usage : /factorial.sh n
fact() {
    n=$1
    if [ "$n" -le 1 ]; then
        echo 1
        return
    fi
    res=1
    for ((i=2; i<=n; i++)) do
        res=$((res * i))
    done
    echo "fact"
    if [ $# -ne 1 ]; then
        echo "Usage : factorial - negative integer"
        exit 1
    fi
    for arg in $@; do
        if ! [[ $arg =~ ^[0-9]+[.]? ]]; then
            echo "$arg" not a non-neg. integer skipping"
            continue
        fi
        echo "$arg! = $(fact "$arg")"
    done
}
```