8. Test

test□ □□ □□ □ □□□□□□ □□□□□ . □□□□ □□ □□□□ □□□ □□ □□ □□□ □□□ □□□ □□ □□ □□□□ . test□ []□ □ □□ □□□□□ . []□ □ □□□□□ □ □□ □□ □□□ □□ □□□□ □□ □□□ □□□□□ . □□ □□□□□ □□ □□□□ □□□□ (□ , Unix □□□ □□□ □□□□ □□□□ □ □□□□ [□ □□□□ □□□ □□□□□):

```
$ type [
[ is a shell builtin
$ which [
/usr/bin/[
$ ls -l /usr/bin/[
lrwxrwxrwx 1 root root 4 Mar 27 2000 /usr/bin/[ -> test
```

□ , '['□ □□□□ ls □ □□ □□□□□ □□□□□ □□□□□□□ □□□□ □□□□ □□□ □□□ :

```
if [$foo = "bar" ]
```

□ □□□□ □□□ , □□ '['□ □□ ']'□ test$foo = "bar"]□□ □□□□□ . □□ □□□ □□□ □□□ □□□□ . □□ □□□ 'SPACE'□□ □□□ □□ □□□□□ , □□□ □□□ □□□□ □□□□ 'SPACE'□ □□ □□□□ □□□□ :

```
if SPACE [ SPACE "$foo" SPACE = SPACE "bar" SPACE ]
```

□□ : □□ □□ □□□ □□□ "=="□ □□□□□ , □□ □□□□ □□□□ □□□□□ □□ "="□ □□□□□ □□□ □□ "-eq"□ □□□□ □□□ .

test□ □□□□□ □□□ □□ □□□□□□□ . □□□ □□□ □□□□□ "man test"□ □□□□□ . □□□ □□□□ □ □□ □□□□ □□□□ □□□ □□□□□ .

test□ if □ while □□ □□ □□□□□ □□□□ □□□ □□ □□□□ . test□□ □□□□□ □□□□ □□□□□ □□ □□□□ □□ □ □ □□ □□□ □□□□ □□□ □□□□ □□ □□ □□□□□ □□□ ! if...then...else...□ □□□ □□□ □□□□ :

```
if [ ... ] then
  # if-code
else
  # else-code
fi
```

fi□ □□□ □ □□□□ □□ □□□□□ !□□□ □□□□ □ esac□ □□ □□ □□□□□ . □□ □□□ □□□□□ . "if [...]"□ "then" □□□ □□ □□ □□ □□□ □□□ . □□ □□□□ ";"□□ □□□ □□ □□□□ :

```
if [ ... ]; then
  # do something
fi
```

□□□ □□ elif□ □□□ □□ □□□□ :

```
if  [ something ]; then
 echo "Something"
 elif [ something_else ]; then
   echo "Something else"
 else
   echo "None of the above"
fi
```

[something] □□□□ □□□□ "Something"□ □□□□ , □□□ □□□ [something_else]□ □□□□□□ .
[something_else]□ □□□□□ □□□□ “Something else"□ □□□□□ . □□ □□ □□□□ □□□□ "None of the above"□ □□□□□ .

□□ □□ □□□□ □□□□ □□ □□ X□ □□□ □□□ □□□ □□□ (-1, 0, 1, hello, bye □ □□□ □□□). □□□ □□ □□ □□□ (□□ - 1□□□ □□□ □□□□ □□□ □□ □□□ □□ Dave□□ □□□□□□):

```
$ X=5
$ export X
$ ./test.sh
  ... output of test.sh ...
$ X=hello
$ ./test.sh
  ... output of test.sh ...
$ X=test.sh
$ ./test.sh
  ... output of test.sh ...
```

□□ □□ $X□ □□ □□ □□ (□ : /etc/hosts)□□ □□□□ □□ □□□□□ .

```
#!/bin/sh
if [ "$X" -lt "0" ]
```

```
then
  echo "X is less than zero"
fi
if [ "$X" -gt "0" ]; then
  echo "X is more than zero"
fi
[ "$X" -le "0" ] && \
      echo "X is less than or equal to  zero"
[ "$X" -ge "0" ] && \
      echo "X is more than or equal to zero"
[ "$X" = "0" ] && \
      echo "X is the string or number \"0\""
[ "$X" = "hello" ] && \
      echo "X matches the string \"hello\""
[ "$X" != "hello" ] && \
      echo "X is not the string \"hello\""
[ -n "$X" ] && \
      echo "X is of nonzero length"
[ -f "$X" ] && \
      echo "X is the path of a real file" || \
      echo "No such file: $X"
[ -x "$X" ] && \
      echo "X is the path of an executable file"
[ "$X" -nt "/etc/passwd" ] && \
echo "X is a file which is newer than /etc/passwd"
```

□□□□ (;)□ □□□□ □ □□ □□□ □ □□□ □□ □□□□□ . □□ □□□ if □□□ □□□ □□□ □□□□ □□ □□ □□□□□ . □□□□□ □□□ □□ □□□ □□ □□ □□□ □ □ (□□ □ □□)□ □□□ □□□□ □□ □□□□□ . □□ □□□□ □□□ □ □□□□□ . □□ □□ □□□□ □□ □□□□□□ .

□ □□□□ □ □ □□□ test□ □□ , □□□ □ □□ □□□ □□ □□ □□□□ □□□ □ □□□□ .

-a, -e(□ □ "□□□ □□□ "□ □□), -S(□□□ □□), -nt(□□□ □□ □□ □□), -ot(□□□ □□ □□□ □□), -ef(□□□ □□□ □□□ □□) □ -O(□□□ □ □□□ □□)□ □□ □ □ (□ : Solaris, AIX, HPUX □□ /bin/sh)□□ □□□ □ □□□ □□ □□□ □□ Aaron□□ □□□□□□ .

if □□ □□□□ □ □□□ □□□ □□□□ : && □ || □□□ □□ □□□ □ □□ □□□ □ □□□ □□□ □□□□□ .

```
#!/bin/sh
[ $X -ne 0 ] && echo "X isn't zero" || echo "X is zero"
[ -f  $X ] && echo "X is a file" || echo "X is not a file"
[ -n  $X ] && echo "X is of non-zero length" || \
      echo "X is of zero length"
```

□ □ □ □ [□ □ (□ □ □)□ □ □ , □ □ test□ □ . □ □ □ □ □ □ □ □ □ □ □ . if...then...else... □ □ □ □ □ . □ □ □ □ □ □ □ □ □ □ [...] □ □ □ □ .

X□ □ □ □ □ □ □ □ □ □ □ :

```
test.sh: [: integer expression expected before -lt
test.sh: [: integer expression expected before -gt
test.sh: [: integer expression expected before -le
test.sh: [: integer expression expected before -ge
```

□ -lt, -gt, -le □ -ge □ □ □ □ □ □ □ □ . !=□ □ □ □ "5"□ □ □ □ "Hello"□ □ □ □ □ □ □ □ □ . □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ :

```
echo -en "Please guess the magic number: "
read X
echo $X | grep "[^0-9]" > /dev/null 2>&1
if [ "$?" -eq "0" ]; then
  # If the grep found something other than 0-9
  # then it's not an integer.
  echo "Sorry, wanted a number"
else
  # The grep found only 0-9, so it's an integer.
  # We can safely do a test on it.
  if [ "$X" = "7" ]; then
    echo "You entered the magic number!"
  fi
fi
```

□ □ □ □ □ □ □ □ □ □ □ □ . □ □ '□ - 2□ '(10□)□ □ , grep□ □ □ □ □ (0~9)□ □ □ □ □ □ □ grep [^0~9]□ □ (^)□ □ □ □ □ □ . □ □ □ □ (□ □ □ □ □ □)□ □ □ □ . □ ? >/dev/ null 2>&1□ □ □ □ □ □ □ □ □ "null" □ □ . □ □ grep -v [0-9]□ □ □ □ □ □ .

□□□ □□ while □□□□ test□ □□□ □ □□□□ :

```
#!/bin/sh
X=0
while [ -n "$X" ]
do
  echo "Enter some text (RETURN to quit)"
  read X
  echo "You said: $X"
done
```

□ □□□ RETURN□ □□ □□□ □□ □□□ □□□□□ (X□ □□□ 0□□□). □□□□□ □□□□ □□□□ □□□ □□ Justin Heath□□ □□□□□□ . □ □□ [-n "$X"]□□ $X □□□ □□□□ □□□ □□□□□ . □ □□□□ □□□ $X□ □□ □□ □ □□□□ □□ □□□□ . □□□□ □□□ □ □□□□□ □□□□ □□□□□ □□□□□ □□ □□□□□□ :

```
$ ./test2.sh
Enter some text (RETURN to quit)
fred
You said: fred
Enter some text (RETURN to quit)
wilma
You said: wilma
Enter some text (RETURN to quit)
```

□□□ □□□ :

```
$
```

□□ □□ □□ □□□□ □ □□□ □□□ □ □□□□ :

```
#!/bin/sh
X=0
while [ -n "$X" ]
do
  echo "Enter some text (RETURN to quit)"
  read X
  if [ -n "$X" ]; then
    echo "You said: $X"
  fi
done
```

[illegible] [illegible] if [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] . [illegible] [illegible] :

```
if [ "$X" -lt "0" ]

then

  echo "X is less than zero"

fi
```

.... [illegible] ....

```
if [ ! -n "$X" ]; then

  echo "You said: $X"

fi
```

if [illegible] then [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] . [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] if [illegible] then [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] . [illegible] [illegible] [illegible] [illegible] [illegible] :

```
if [ ! -n "$X" ]

  echo "You said: $X"
```

[illegible] then[illegible] fi[illegible] [illegible] [illegible] .

Revision #3
Created 16 January 2024 13:25:25 by Enigma
Updated 16 January 2024 13:48:11 by Enigma