

1. Hello world 출력

- Echo를 활용하여 문자열 hello world를 출력한다

```
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ls
README.md  lab2-0.sh
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-0.sh
hello world
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$
```

2. 입력받은 수 만큼 hello world출력

- 입력받은 파라미터 = \$1 / 이를 변수에 할당하여 그 수만큼 반복

```
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-1.sh 6
6
hello world
hello world
hello world
hello world
hello world
hello world
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-1.sh 1
1
hello world
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$
```

3. 입력받은 두 수의 차를 출력하기

- Num1=1번 파라미터 / num2=2번 파라미터를 할당 받고 두 수의 차를 출력한다

```
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-2.sh 3 1
2
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-2.sh 1 3
-2
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$
```

4. bmi계산하기

- 소수점 2자리까지 활용하기 위해 scale=2를 사용
- 나누셈을 위해 bc(basic calculator)사용
- If 구문에선 bmi가 18.5보다 작다는 것이 참이면 (=> 1 -eq ~)처럼 활용

```

kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ls
DB.txt      lab2-0.sh  lab2-2.sh  lab2-4.sh  lab2-6.sh  lab2-8.sh
README.md   lab2-1.sh  lab2-3.sh  lab2-5.sh  lab2-7.sh  lab2-9.sh
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-3.sh 50 170
17.30
저체중
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-3.sh 65 170
22.49
정상입니다
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-3.sh 85 170
29.41
과체중
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ _

```

5. case문 활용하기

- 수업자료 17p를 활용 / 변형하여 사용

```

kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-4.sh
리눅스가 재미있나요?(yes/no)
yes
yes
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-4.sh
리눅스가 재미있나요?(yes/no)
no
no
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-4.sh
리눅스가 재미있나요?(yes/no)
hmm
yes or no로만 입력
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$

```

6. 내부함수 사용하기

- 내부 함수를 생성
- 내부 함수에서 입력 받은 명령어를 전달받아 eval로 수행

```

kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-5.sh ls
프로그램 시작
함수 안
README.md lab2-0.sh lab2-1.sh lab2-2.sh lab2-3.sh lab2-4.sh lab2-5.sh
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$

```

7. 입력된 폴더가 있는 확인. 압축 및 해제를 수행.

- 입력된 이름의 디렉토리가 존재하는지 확인
- 없으면 eval을 통해 디렉토리 생성 및 파일 생성
- 해당 파일들 압축한 tar만든 후 새 디렉토리를 생성하여 mv

- 새로만든 디렉토리에서 압축해제

```
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ls
README.md lab2-0.sh lab2-1.sh lab2-2.sh lab2-3.sh lab2-4.sh lab2-5.sh lab2-6.sh
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-6.sh hello
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ls
README.md hello lab2-0.sh lab2-1.sh lab2-2.sh lab2-3.sh lab2-4.sh lab2-5.sh lab2-6.sh
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ cd hello/
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script/hello$ ls
hello hello1.txt hello2.txt hello3.txt hello4.txt hello5.txt
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script/hello$ cd hello/
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script/hello/hello$ ls
hello.tar hello1.txt hello2.txt hello3.txt hello4.txt hello5.txt
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script/hello/hello$
```

8. 심볼릭 링크

- 반복문과 eval을 활용하여 파일/디렉토리를 만든 각 폴더에 해당 파일을 링크한다

```
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ls
README.md lab2-0.sh lab2-1.sh lab2-2.sh lab2-3.sh lab2-4.sh lab2-5.sh lab2-6.sh lab2-7.sh
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ vi lab2-7
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ vi lab2-7.sh
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-7.sh hello
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ls
README.md hello1.txt hello2.txt hello3.txt hello4.txt hello5.txt lab2-1.sh lab2-3.sh lab2-5.sh lab2-7.sh
hello1 hello2 hello3 hello4 hello5 lab2-0.sh lab2-2.sh lab2-4.sh lab2-6.sh
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ cd hello1
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script/hello1$ ls
hello1.txt
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script/hello1$ ll
total 8
drwxr-xr-x 2 kiseok kiseok 4096 Nov  8 12:09 ./
drwxr-xr-x 8 kiseok kiseok 4096 Nov  8 12:09 ../
lrwxrwxrwx 1 kiseok kiseok  10 Nov  8 12:09 hello1.txt -> hello1.txt
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script/hello1$
```

9. 이름과 생일 또는 전화번호를 DB.txt에 기록

- DB.txt파일이 존재하는지 확인 / 파라미터로 이름 전화번호가 전달되고
- 파일이 존재한다면 >>를 통해 전달받은 파라미터를 맨 뒤에 이어서 작성 / 없다면 >을 통해 새로 파일 만든 후 작성

```
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ls
README.md lab2-0.sh lab2-1.sh lab2-2.sh lab2-3.sh lab2-4.sh lab2-5.sh lab2-6.sh lab2-7.sh lab2-8.sh
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-8.sh nam 010-3923-8981
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ls
DB.txt README.md lab2-0.sh lab2-1.sh lab2-2.sh lab2-3.sh lab2-4.sh lab2-5.sh lab2-6.sh lab2-7.sh lab2-8.sh
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ cat DB.txt
nam 010-3923-8981
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-8.sh ki 010-3923-8982
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ cat DB.txt
nam 010-3923-8981
ki 010-3923-8982
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$
```

10. db에서 이름으로 검색하기

- grep을 활용하여 DB.txt에서 해당 파라미터를 포함한 라인을 출력한다

```
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ls
DB.txt      lab2-0.sh  lab2-2.sh  lab2-4.sh  lab2-6.sh  lab2-8.sh
README.md   lab2-1.sh  lab2-3.sh  lab2-5.sh  lab2-7.sh  lab2-9.sh
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ vi lab2-9.sh
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-9.sh nam
nam 010-3923-8981
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-9.sh na
nam 010-3923-8981
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-9.sh n
nam 010-3923-8981
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-9.sh k
ki 010-3923-8982
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$ ./lab2-9.sh ki
ki 010-3923-8982
kiseok@DESKTOP-MAU5RL6:/home/shell-script/shell-script$
```