

Örnek-1

```
#!/bin/bash

# Masaüstü dizinini belirleyin
DESKTOP_DIR=~/Desktop

# Dosya adını belirleyin
FILE_NAME="mmtf"

# Tam dosya yolunu oluşturun
FILE_PATH="$DESKTOP_DIR/$FILE_NAME"

# Dosya oluştur ve kullanıcıdan bilgi al
echo "Dosyaya yazılacak bilgiyi girin (Ctrl+D ile bitirin):"
cat > "$FILE_PATH"

# Dosyayı başarıyla kaydettiğimizi bildirin
echo "Bilgi başarıyla $FILE_PATH dosyasına kaydedildi."

# Kullanıcıdan arama terimini al
read -p "Dosya içinde aramak istediğiniz terimi girin: " search_term

# Arama yap ve sonucu ekrana yazdır
echo "Arama sonuçları:"

grep "$search_term" "$FILE_PATH"

# Arama sonucunun olmadığı durumu ele al
if [ $? -ne 0 ]; then
    echo "Arama terimi \"$search_term\" dosyada bulunamadı."
fi
```

Açıklama

- Masaüstü Dizinini Belirleme:** `DESKTOP_DIR=~/Desktop` ifadesi, kullanıcının masaüstü dizinini temsil eder. `~` işareti, kullanıcının ana dizinine işaret eder.
- Dosya Adı ve Yolunu Belirleme:** `FILE_NAME` değişkeni dosya adını tutar ve `FILE_PATH` değişkeni tam dosya yolunu oluşturur.
- Dosya Oluşturma ve Bilgi Alma:**

- `echo "Dosyaya yazılacak bilgiyi girin (Ctrl+D ile bitirin):"` komutu kullanıcıya bilgi girmesi gerektiğini söyler.
 - `cat > "$FILE_PATH"` komutu, kullanıcının girdiği bilgiyi belirtilen dosyaya kaydeder. Kullanıcı, girişini tamamladığında `Ctrl+D` tuş kombinasyonunu kullanarak girişi sonlandırır.
4. **Arama Terimini Alma:** `read -p "Dosya içinde aramak istediğiniz terimi girin: " search_term` komutu kullanıcıdan bir arama terimi alır.
 5. **Dosyada Arama Yapma:** `grep "$search_term" "$FILE_PATH"` komutu, belirtilen terimi dosya içinde arar ve sonuçları ekrana yazdırır.
 6. **Arama Sonucu Kontrolü:**
 - `if [$? -ne 0]; then` ifadesi, `grep` komutunun çıkış durumunu kontrol eder. Eğer `grep` komutu herhangi bir sonuç bulamazsa, çıkış durumu sıfırdan farklı olur.
 - `echo "Arama terimi \"$search_term\" dosyada bulunamadı."` komutu, arama teriminin dosyada bulunamadığını bildirir.

Örnek-2

```
#!/bin/bash

# Dosya adı ve yolu
FILE_NAME="user_info.txt"
FILE_PATH="$HOME/$FILE_NAME"

# Yeni dosya oluştur veya mevcut dosyayı temizle
> "$FILE_PATH"

# Kullanıcıdan bilgiler al ve dosyaya yaz
while true; do
    echo "Kullanıcı Bilgileri Girişi"
    read -p "Ad: " name
    read -p "Yaş: " age
    read -p "Meslek: " job
    echo "Ad: $name, Yaş: $age, Meslek: $job" >> "$FILE_PATH"
    read -p "Başka bir kullanıcı eklemek istiyor musunuz? (E/H): " choice
    if [[ "$choice" =~ ^[Hh]$ ]]; then
        break
    fi
done

# Dosyayı ekrana yazdır
echo "Kullanıcı Bilgileri:"
cat "$FILE_PATH"

# Kullanıcıdan arama terimi al ve dosyada ara
read -p "Aramak istediğiniz meslek: " search_term
echo "Arama sonuçları:"
grep "Meslek: $search_term" "$FILE_PATH"
if [ $? -ne 0 ]; then
    echo "\"$search_term\" mesleği ile ilgili kayıt bulunamadı."
fi

# Dosyayı sıralı şekilde ekrana yazdır
echo "Kullanıcı bilgileri alfabetik sıraya göre:"
```

```
sort "$FILE_PATH"

# Dosya içeriğini ters sıraya göre ekrana yazdır

echo "Kullanıcı bilgileri ters alfabetik sıraya göre:"

sort -r "$FILE_PATH"
```

Açıklama

1. Dosya Adı ve Yolu:

- `FILE_NAME="user_info.txt"` ve `FILE_PATH="$HOME/$FILE_NAME"` değişkenleri, kullanıcı bilgilerinin kaydedileceği dosyanın adını ve yolunu belirtir. `$HOME` kullanıcı ana dizinini temsil eder.

2. Dosya Oluşturma veya Temizleme:

- `> "$FILE_PATH"` komutu, belirtilen dosyayı oluşturur veya mevcut dosyayı temizler.

3. Bilgi Girişi ve Dosyaya Yazma:

- `while true; do ... done` döngüsü, kullanıcı bilgilerini tekrar tekrar girmesine izin verir.
- `read -p "Ad: " name, read -p "Yaş: " age` ve `read -p "Meslek: " job` komutları, kullanıcıdan adı, yaşı ve mesleği alır.
- `echo "Ad: $name, Yaş: $age, Meslek: $job" >> "$FILE_PATH"` komutu, alınan bilgileri dosyaya ekler.
- `read -p "Başka bir kullanıcı eklemek istiyor musunuz? (E/H): " choice` komutu, kullanıcıya daha fazla bilgi eklemek isteyip istemediğini sorar. Kullanıcı `H` veya `h` girerse döngü sonlanır.

4. Dosya İçeriğini Ekrana Yazdırma:

- `cat "$FILE_PATH"` komutu, dosya içeriğini ekrana yazdırır.

5. Arama İşlemi:

- `read -p "Aramak istediğiniz meslek: " search_term` komutu, kullanıcıdan bir meslek arama terimi alır.
- `grep "Meslek: $search_term" "$FILE_PATH"` komutu, dosyada belirtilen mesleği arar ve sonuçları ekrana yazdırır. Eğer meslek bulunamazsa, `"\"$search_term\" mesleği ile ilgili kayıt bulunamadı."` mesajı gösterilir.

6. Sıralama İşlemleri:

- `sort "$FILE_PATH"` komutu, dosya içeriğini alfabetik sıraya göre sıralar ve ekrana yazdırır.
- `sort -r "$FILE_PATH"` komutu, dosya içeriğini ters alfabetik sıraya göre sıralar ve ekrana yazdırır.

Örnek-3

```
#!/bin/bash

# Kullanıcıdan öğrenci notlarını al ve ortalamaları hesapla

declare -A students

while true; do

    read -p "Öğrenci adı (bitirmek için 'exit' yazın): " name

    if [ "$name" == "exit" ]; then

        break

    fi

    read -p "$name'in notlarını virgülle ayrılmış olarak girin (örnek: 85,90,78): " grades

    students["$name"]=$grades

done

echo "Öğrenci Notları ve Ortalamaları:"

for name in "${!students[@]}"; do

    IFS=',' read -r -a grade_array <<< "${students[$name]}"

    total=0

    count=0

    for grade in "${grade_array[@]}"; do

        total=$((total + grade))

        count=$((count + 1))

    done

    average=$((echo "scale=2; $total / $count" | bc))

    echo "$name: Ortalama Notu = $average"

done

read -p "Not eşliğini girin: " threshold

echo "Not eşliğinin altında kalan öğrenciler:"

for name in "${!students[@]}"; do

    IFS=',' read -r -a grade_array <<< "${students[$name]}"

    total=0

    count=0

    for grade in "${grade_array[@]}"; do
```

```
total=$((total + grade))
count=$((count + 1))
done
average=$(echo "scale=2; $total / $count" | bc)
if (( $(echo "$average < $threshold" | bc -l) )); then
    echo "$name: Ortalama Notu = $average"
fi
done
```