

Számítógép architektúrák

BSc

9. Gyak.

2023. 12. 06.

Készítette:

Balogh Martin Bsc

Programtervező inf..

OIKGCW

Miskolc, 2023

1

```
#!/bin/bash

echo "1-10-ig számok kiírása"
echo "For ciklussal: "
for ((i=1;i<11;i++)); do
echo -n "$i "
done

echo -e "\nWhile ciklussal: "
i=1
while [[ $i -le 10 ]]; do
echo -n "$i "
((i++))
done

echo -e "\nUntil ciklussal: "
i=1
until [[ $i -gt 10 ]]; do
echo -n "$i "
((i++))
done

echo -e "\n"
```

2

```
#!/bin/bash

# Véletlen szám generálása 1 és 100 között
random_number=$((RANDOM % 100 + 1))

echo "Véletlen szám: $random_number"
echo "Nála kisebb négyzetszámok:"

for (( i=1; i<$random_number; i++ )); do
    square=$((i * i))
    if (( square < random_number )); then
        echo $square
    fi
done
```

3

```
#!/bin/bash

# Ellenőrzi, hogy a bemeneti paraméter meg van-e adva
if [ -z "$1" ]; then
    echo "Hiba: Adj meg egy pozitív számot bemeneti paraméterként."
    exit 1
fi

# Ellenőrzi, hogy a bemeneti paraméter pozitív egész szám-e
if ! [[ "$1" =~ ^[1-9][0-9]*$ ]]; then
    echo "Hiba: A bemeneti paraméternek pozitív egész számnak kell lennie."
    exit 1
fi

# A prímszám-ellenőrzés függvénye
is_prime() {
    num=$1
    if [ $num -lt 2 ]; then
        return 1 # Nem prím
    fi

    for ((i=2; i*i<=$num; i++)); do
        if [ $((num % i)) -eq 0 ]; then
            return 1 # Nem prím
        fi
    done

    return 0 # Prím
}

is_prime "$1"

if [ $? -eq 0 ]; then
    echo "$1 prím szám "
```

4

```
#!/bin/bash

sum_numbers() {
    local sum=0

    if [ "$#" -eq 0 ]; then
        echo 0
        return
    fi

    for arg in "$@"; do
        sum=$((sum + arg))
    done

    echo $sum
}

result=$(sum_numbers 3 5 7)
echo "Eredmény: $result"

# Függvény hívása argumentumok nélkül
result=$(sum_numbers)
echo "Eredmény: $result"
```

5

```
#!/bin/bash

generate_random_numbers() {
    local N=${1:-5} # Alapértelmezett érték: 5
    local x=${2:-1}  # Alapértelmezett érték: 1
    local y=${3:-90} # Alapértelmezett érték: 90

    for ((i=0; i<N; i++)); do
        random_number=$((RANDOM % (y - x + 1) + x))
        echo $random_number
    done
}

# Függvény hívása 10 véletlen számmal 800 és 900 között
echo "Véletlen számok 800 és 900 között:"
generate_random_numbers 10 800 900

# Függvény hívása 15 véletlen számmal -10 és 10 között
echo "Véletlen számok -10 és 10 között:"
generate_random_numbers 15 -10 10
```

6

```
#!/bin/bash

array=()

echo "A tömb random számokkal: "
for((i=0;i<10;i++)); do
    array[i]=$((RANDOM % 100 + 1))
    echo -n "${array[i]} "
done

echo -e "\nA tömb értékei megnövelve 1-el: "

for((i=0;i<10;i++)); do
    array[i]=$((array[i] + 1))
    echo -n "${array[i]} "
done

echo -e "\n"
```



```
#!/bin/bash

read -p "Adja meg a tömb méretét (pozitív egész szám): " N
# Ellenőrzés, hogy N pozitív egész szám-e
if ! [[ "$N" =~ ^[1-9][0-9]*$ ]]; then
    echo "Hiba: N-nek pozitív egész számnak kell lennie."
    exit 1
fi

# Tömb létrehozása és feltöltése véletlen számokkal
array=()
for ((i=0; i<N; i++)); do
    array[i]=$((RANDOM % 100 + 1))
done

# Minimum elem keresése
min_index=0
for ((i=1; i<N; i++)); do
    if [ "${array[i]}" -lt "${array[min_index]}" ]; then
        min_index=$i
    fi
done

# Maximum elem keresése
max_index=0
for ((i=1; i<N; i++)); do
    if [ "${array[i]}" -gt "${array[max_index]}" ]; then
        max_index=$i
    fi
done

# Maximum elem keresése
max_index=0
for ((i=1; i<N; i++)); do
    if [ "${array[i]}" -gt "${array[max_index]}" ]; then
        max_index=$i
    fi
done

# Elemek összege és átlaga
sum=0
for ((i=0; i<N; i++)); do
    sum=$((sum + array[i]))
done
average=$((sum / N))

echo "Tömb elemei: ${array[@]}"
echo "Minimum: $((min_index + 1)). elem, ${array[min_index]}"
echo "Maximum: $((max_index + 1)). elem, ${array[max_index]}"
echo "Elemek összege: ${sum}"
echo "Elemek átlaga: ${average}"
```