

# **Számítógép architektúrák**

## **BSc**

**9. Gyak.**  
**2023. 12. 09.**

**Készítette:**

Dongó Petra Bsc

Szak: programtervező informatikus

Neptunkód: A3U8XV

**Miskolc, 2023**

**1. feladat - Írjon ki 0-tól 10-ig a számokat, for, while i=0, until ciklussal is!**

```
$ A3U8XV_9.gyak1.sh
1  #!/bin/bash
2
3  for ((i=0; i<=10; i++)); do
4      echo $i
5  done
6  counter=0
7  while [ $counter -le 10 ]; do
8      echo $counter
9      ((counter++))
10 done
11 counter=0
12 until [ $counter -gt 10 ]; do
13     echo $counter
14     ((counter++))
15 done
```

**2. feladat - Generáljon egy véletlen számot és írja ki a nála kisebb négyzetszámokat! A véletlenszám generálás Bashben a RANDOM változón keresztül elérhető.**

```
$ A3U8XV_9.gyak2.sh
1  #!/bin/bash
2
3  random_number=$((RANDOM % (100 +1 ))
4  printf "A véletlenszerűen generált szám: " "$random_number"
5  for ((i = 0; i*i < random_number; i++)); do
6      square=$((i i))
7      echo $square
8  done
9  exit 0
```

3. feladat - Írjon egy shell scriptet, ami bemeneti paraméterként egyetlen pozitív számot vár (hibát ír, ha nem ezt kap). A program feladata, hogy kiírja, hogy a kapott szám prím-e.

```
$ A3U8XV_9.gyak3.sh
1  #!/bin/bash
2
3  printf "Kérek egy pozitív egész számot: " pozitiveszam
4  if [[ $pozitiveszam !=~ ^[1-9][0-9]*$ ]]; then
5      printf "Ez nem pozitív egész szám."
6  fi
7  if [[ $pozitiveszam -lt 2 ]]; then
8      printf "Ez nem prím szám."
9      exit 1
10 fi
11 for ((i = 2; i*i <= pozitiveszam; i++)); do
12     if [ $((pozitiveszam % i)) -eq 0 ]; then
13         printf "Ez nem prím szám."
14         exit 1
15     fi
16 done
17 printf "Ez prím szám"
18 exit 0
```

4. feladat - Töltsön fel egy 10 elemű tömböt véletlen számokkal, majd menjen végig a számokon és növelje meg őket 1-gyel!

```
$ A3U8XV_9.gyak4.sh
1  #!/bin/bash
2
3  generate_random_number() {
4      echo $((RANDOM % 100))
5  }
6  for ((i = 0; i < 10; i++)); do
7      random_number=$(generate_random_number)
8      my_array[$i]=$random_number
9  done
10 echo "Eredeti tömb:"
11 echo "${my_array[@]}"
12 for ((i = 0; i < 10; i++)); do
13     ((my_array[i]++))
14 done
15 echo "Minden elemhez hozzáadtunk 1-et:"
16 echo "${my_array[@]}"
```

5. feladat - Hozzon létre egy N elemű tömböt, ahol N-t read paranccsal kérje be, és ellenőrizze, hogy pozitív egész szám (hibával visszatérünk, ha nem az). Töltse fel véletlen számokkal a tömböt, ahol a véletlen számok az [1-100] intervallumból kerülnek ki! (...)

```
$ tomb2.sh
1  #!/bin/bash
2
3  generate_random_number() {
4      echo $((RANDOM % 101))
5  }
6  printf "Kérek egy pozitív egész számot: " pozitiveszam
7  if [[ $positiveszam !=~ ^[1-9][0-9]*$ ]]; then
8      printf "Ez nem pozitív egész szám."
9  fi
10 for ((i = 0; i < pozitiveszam; i++)); do
11     random_number=$(generate_random_number)
12     my_array[$i]=$random_number
13 done
14 min=$my_array[0]
15 max=$my_array[0]
16 minpoz=0
17 maxpoz=0
18 for ((i = 1; i < pozitiveszam; i++)); do
19     if (($my_array[i] < $min)); then
20         $min=$my_array[i]
21         minpoz=$i
22     fi
23     if (($my_array[i] > $max)); then
24         $max=$my_array[i]
25         maxpoz=$i
26     fi
27 done
28 osszeg=0
29 for ((i = 0; i < pozitiveszam; i++)); do
30     osszeg=$my_array[i]+$osszeg
31 done
32 atlag=$osszeg/$positiveszam
33 exit 0
34
```

6. feladat - Készítsen egy függvényt, ami összeadja az argumentumban megadott számokat. Ha nem kap argumentumot, akkor 0-t ír ki, egyébként az argumentumok összegét. Az argumentumokat nem szükséges ellenőrizni, hogy számok-e.

```
$ fg1.sh
1  #!/bin/bash
2
3  if [ $# -eq 0 ]; then
4      echo "Összeg: 0"
5  else
6      osszeg=0
7      for arg in "$@"; do
8          osszeg=$((osszeg + arg))
9      done
10     echo "Összeg: $osszeg"
11 fi
12 exit 0
```

7. feladat - Írjon függvényt, ami legenerál N darab véletlen számot egy [x-y] intervallumban. N, x és y értékeit paraméterekként kapja meg a függvény. (...)

```
$ fg2.sh
1  #!/bin/bash
2
3  generate_random_number() {
4      local N=${1:-5}
5      local x=${2:-1}
6      local y=${3:-90}
7      for ((i = 0; i < N; i++)); do
8          random_szam=$((RANDOM % (y - x + 1) + x))
9          echo $random_szam
10     done
11 }
12 generate_random_number 10 800 900
13 generate_random_number 15 -10 10
14 exit 0
```