

Számítógép Architektúrák

BSc

9. Gyak.

2023. 12. 06.

Készítette:

Sebe Zsolt BSc

Programtervező Informatikus

ACC02G

Miskolc, 2023

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

```
1 #!/bin/bash
2
3
4 echo 'for ciklus:'
5
6 for i in $(seq 0 10); do
7     echo $i
8 done
9
10 echo 'while ciklus:'
11
12 declare -i i=0
13 while [ $i -le 10 ]; do
14     echo $i
15     i=$((i + 1))
16 done
17
18 echo 'until ciklus:'
19
20 declare -i i=0
21 until [ $i -gt 10 ]; do
22     echo $i
23     i=$((i + 1))
24 done
```

```
acc02g@acc02g-pc $ ./ciklus1.sh
for ciklus:
0
1
2
3
4
5
6
7
8
9
10
while ciklus:
0
1
2
3
4
5
6
7
8
9
10
until ciklus:
0
1
2
3
4
5
6
7
8
9
10
acc02g@acc02g-pc $
```

2. feladat – Generáljon egy véletlen számot és írja ki a nála kisebb négyzetszámokat!

```
1 #!/bin/bash
2
3 declare -ri n=$RANDOM
4
5 echo "A random szán: $n"
6
7 declare -i i=0
8 declare -i square_i=0
9 while [ $square_i -lt $n ]; do
10     echo $square_i
11     i=$((i + 1))
12     square_i=$((i * i))
13 done
```

```
acc02g@acc02g-pc $ ./ciklus2.sh
A random szán: 6358
0
1
4
9
16
25
36
49
64
81
100
121
144
169
196
225
256
```

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.

```
1  #!/bin/bash
2
3  declare -i n
4  read -rp 'Adj meg egy pozitív számot: ' n
5  declare -r n
6
7  if [ $n -lt 1 ]; then
8      echo "$n nem pozitív egész szám"
9      exit 1
10 fi
11
12 if [ $n -eq 1 ]; then
13     echo '1 nem prím szám'
14     exit
15 fi
16
17 declare -i i=2
18 while [ $i -lt $n ]; do
19     if [ $((n % i)) -eq 0 ]; then
20         echo "$n nem prím szám"
21         exit
22     fi
23
24     i=$((i + 1))
25 done
26
27 echo "$n prím szám"
```

```
acc02g@acc02g-pc $ ./ciklus3.sh
Adj meg egy pozitív számot: 4
4 nem prím szám
acc02g@acc02g-pc $
```

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

```
1 #!/bin/bash
2
3 declare -a rand_nums
4
5 for i in $(seq 0 9); do
6     rand_nums+=($RANDOM)
7 done
8
9 echo 'Eredeti számok:'
10
11 for i in "${rand_nums[@]}"; do
12     echo $i
13 done
14
15 for i in $(seq 0 9); do
16     rand_nums[i]=$((rand_nums[i] + 1))
17 done
18
19 echo 'Új számok:'
20
21 for i in "${rand_nums[@]}"; do
22     echo $i
23 done
```

```
acc02g@acc02g-pc $ ./tomb1.sh
Eredeti számok:
28125
1694
1253
7079
19029
32391
27516
18045
8970
784
Új számok:
28126
1695
1254
7080
19030
32392
27517
18046
8971
785
acc02g@acc02g-pc $
```

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!

Ezután végezze el a következő műveleteket a tömbre:

- Minimum elem kiírása (hányadik elem és mi az értéke).
- Maximum elem kiírása (hányadik elem és mi az értéke).
- Írja ki az elemek összegét.
- Írja ki az elemek átlagát.

```
1  #!/bin/bash
2
3  declare -i n
4  read -rp 'Adt meg N-t: ' n
5
6  if [ "$n" -lt 1 ]; then
7      echo "$n nem pozitív egész szám"
8      exit 1
9  fi
10
11 declare -a arr
12
13 for _ in $(seq 1 $n); do
14     arr+=($(RANDOM % 100 + 1))
15 done
16
17 for i in "${arr[@]}"; do
18     echo $i
19 done
20
21 declare -i min=0
22 for i in $(seq 0 $((n - 1))); do
23     if [ "${arr[$i]}" -lt "${arr[$min]}" ]; then
24         min=$i
25     fi
26 done
27 echo "Minimum elem: ${arr[$min]}"
28
29 declare -i max=0
30 for i in $(seq 0 $((n - 1))); do
31     if [ "${arr[$i]}" -gt "${arr[$max]}" ]; then
32         max=$i
33     fi
34 done
35 echo "Maximum elem: ${arr[$max]}"
36
37 declare -i sum=0
38 for i in $(seq 0 $((n - 1))); do
39     sum=$((sum + arr[i]))
```

```
40 done
41 echo "Összeg: $sum"
42
43 avg=$((sum / n))
44 echo "Átlag: $avg"
```

```
acc02g@acc02g-pc $ ./tomb2.sh
Adt meg N-t: 4
85
97
14
56
Minimum elem: 14
Maximum elem: 97
Összeg: 252
Átlag: 63
acc02g@acc02g-pc $
```

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.

```
1 #!/bin/bash
2
3 function sum {
4     declare -i sum=0
5
6     for n; do
7         sum=$((sum + n))
8     done
9
10    echo $sum
11 }
12
13 sum "$@"
```

```
acc02g@acc02g-pc $ ./fg1.sh 2 4 5
11
acc02g@acc02g-pc $
```


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

Ha bármelyik paramétert nem adják meg, akkor a default értékek legyenek a következők:

N=5, x=1, y=90.

Generáltassunk a függvénnyel 10 véletlen számot 800 és 900 között, majd 15 számot –10 és 10 között!

```
1 #!/bin/bash
2
3 function randgen {
4     declare -ri N=${1:-5}
5     declare -ri x=${2:-1}
6     declare -ri y=${3:-90}
7
8     for _ in $(seq 1 $N); do
9         echo $((RANDOM % (y - x + 1) + x))
10    done
11 }
12
13 randgen 10 800 900
14 randgen 15 -10 10
```

```
acc02g@acc02g-pc $ ./fg2.sh
831
845
869
809
868
871
819
840
839
869
2
9
-9
7
-3
-4
10
-6
10
-9
-3
10
-2
9
10
acc02g@acc02g-pc $
```