

Студент (име и презиме):  
 Број индекса:  
 22. септембар 2025.

Факултет инжењерских наука  
 Практикум из рачунарских алата  
 Колоквијум

**Задаци:**

У свим задацима важи претпоставка да се наредбе/програми/скрипте извршавају у **bash** окружењу. Наредбе које извршавају корисници обавезно започињати са \$, а које извршава суперкорисник са #.

**Задатак 1** – [по 2% за сваки тачан од десет одговора, односно 20% укупно]

- [2%] — Израчунати и исписати вредност израза `godina_upisa + (broj_indekса (mod 600))2` (вредности променљивих унети на основу сопственог броја индекса и године уписа).

1 `$ echo $((2020+$((($((611%600))*2))))` решење \_\_\_\_\_  
 \_\_\_\_\_ решење \_\_\_\_\_

- [2%] — Направити 2 датотеке назване `pra1.txt`, `pra2.txt`. Уписати у сваку од датотека по један ред текста.

1 `$ echo садржај1 > pra1.txt` решење \_\_\_\_\_  
 2 `$ echo садржај2 > pra2.txt` \_\_\_\_\_ решење \_\_\_\_\_

- [2%] — Направити трећу датотеку `pra3.txt` унутар које се налази садржај претходне 2 датотеке.

1 `$ cat pra1.txt pra2.txt > pra3.txt` решење \_\_\_\_\_  
 \_\_\_\_\_ решење \_\_\_\_\_

- [2%] — **Не** користећи се наредбом `cp` копирати садржај датотеке `pra1.txt` у датотеку `копија.txt`.

1 `$ cat pra1.txt > копија.txt` решење \_\_\_\_\_  
 \_\_\_\_\_ решење \_\_\_\_\_

- [2%] — Направити датотеку `fin.txt` и онемогућити читање и промену њеног садржаја за све осим за власника датотеке. Извршавање датотеке забранити за све кориснике.

1 `$ touch fin.txt` решење \_\_\_\_\_  
 2 `$ chmod 600 fin.txt` \_\_\_\_\_ решење \_\_\_\_\_

- [2%] — Избрисати директоријум `стари`. Директоријум `стари` се налази унутар тренутног радног директоријума и није празан.

1 `$ rm -rf стари` решење \_\_\_\_\_  
 \_\_\_\_\_ решење \_\_\_\_\_

- [2%] — Излистати све садржаје унутар тренутног радног директоријума чији назив се завршава великим словом.

1 `$ echo *[:upper:]` решење \_\_\_\_\_  
 \_\_\_\_\_ решење \_\_\_\_\_

- [2%] — Искористити садржај датотеке `улаз` као улаз програма `./prog` користећи се редирекцијом.

1 `$ ./prog < улаз` решење \_\_\_\_\_  
 \_\_\_\_\_ решење \_\_\_\_\_

- [2%] — Искористити садржај датотеке `подаци` као улаз програма `./prog` користећи се тзв. пајп-ом.

1 `$ cat подаци | ./prog` решење \_\_\_\_\_  
 \_\_\_\_\_ решење \_\_\_\_\_

- [2%] — Уносом путем тастатуре забележити поруку коју остављате свету, а онда је сачувати као садржај датотеке `порука.txt`.

1 `$ cat - > порука.txt` решење \_\_\_\_\_  
 \_\_\_\_\_ решење \_\_\_\_\_

## Задатак 2 – [40%]

[20%] — Написати **bash** скрипту која као први аргумент прима операцију (додај, уклони, листај), а као други аргумент назив операнда. За операцију додај, потребно је да скрипта дода назив операнда у датотеку **задаци.txt**. За операцију уклони, потребно је уклонити линије унутар датотеке које садрже назив операнда. За операцију листај, потребно је приказати све линије из датотеке које садрже назив операнда, нумерисано. При извршавању скрипте проверити да ли су унети аргументи исправни, као и да ли постоји датотека **задаци.txt**.

решење

```

1      #!/usr/bin/env bash
2
3      if (($# != 2)); then
4          echo "Потребна су два аргумента."
5          exit 1
6      fi
7
8      operacija="$1"
9      operand="$2"
10     datoteka="задаци.txt"
11
12     case "$operacija" in
13         "додај")
14             if [[ -z "$operand" ]]; then
15                 echo "За операцију 'додај' потребан је назив операнда."
16                 exit 1
17             fi
18             echo "$operand" >> "$datoteka"
19             echo "Операнд '$operand' је додат."
20             ;;
21         "уклони")
22             if [[ -z "$operand" ]]; then
23                 echo "За операцију 'уклони' потребан је назив операнда."
24                 exit 1
25             fi
26             if [[ -f "$datoteka" ]]; then
27                 grep -v "$operand" "$datoteka" > temp && mv temp "$datoteka"
28                 echo "Операнд '$operand' је уклоњен."
29             else
30                 echo "Датотека задаци.txt не постоји."
31             fi
32             ;;
33         "листај")
34             if [[ -f "$datoteka" ]]; then
35                 grep -n "$operand" "$datoteka"
36             else
37                 echo "Датотека задаци.txt не постоји."
38             fi
39             ;;
40         *)
41             echo "Непозната операција. Дозвољене су: додај, уклони, листај."
42             exit 1
43             ;;
44     esac

```

решење

[20%] — Написати **bash** скрипту која симулира једноставну игру "Погоди број". Потребно је да скрипта генерише случајан број између 1 и 20, а затим од корисника очекује да погађа број путем стандардног улаза. За сваки покушај, скрипта исписује кориснику упутства ("већи" или "мањи") све док корисник не погоди број. На крају је потребно приказати колико покушаја је било потребно да корисник погоди број.

решење

```

1  #!/usr/bin/env bash
2
3  # Генерисање случајног броја између 1 и 20
4  random_broj=$((RANDOM % 20 + 1))
5  attempts=0
6
7  echo "Добродошли у игру 'Погоди број'!"
8  echo "Погодите број између 1 и 20."
9
10 while true; do
11     echo -n "Унесите ваш број: "
12     read broj
13
14     ((attempts++))
15
16     # Провера да ли је унос број
17     if ! [[ "$broj" =~ ^[0-9]+$ ]]; then
18         echo "Унесите број!"
19         continue
20     fi
21
22     if [[ $broj -eq $random_broj ]]; then
23         echo "Честитамо! Погодили сте број $random_broj!"
24         echo "Потребно вам је било $attempts покушаја."
25         break
26     elif [[ $broj -lt $random_broj ]]; then
27         echo "Тражени број је већи."
28     else
29         echo "Тражени број је мањи."
30     fi
31 done

```

решење

**Задатак 3 – [20%]**

[20%] — У директоријуму /компанија налази се пет поддиректоријума продаја, маркетинг, развој, рачуноводство и излети.

У компанији ради 5 корисника: **марко**, **ана**, **петар**, **милица** и **стефан**. Направити наведене кориснике и подесити лозинку за сваког од њих. Поред наведених 5 корисника, на систему постоји и корисник **директор** (власник компаније).

Уредити дозволе на систему тако да буде задовољено:

- Сви корисници имају пуне привилегије над садржајем директоријума **излети**.
- Корисник **директор** је власник свих директоријума и има пуне привилегије над њима.
- Корисник **марко** ради у продаји и има пуне привилегије само над директоријумом **продаја**, а право читања над осталим директоријумима.
- Корисник **ана** ради у маркетингу и има пуне привилегије само над директоријумом **маркетинг**, а право читања над осталим директоријумима.
- Корисник **петар** ради у развоју и има пуне привилегије само над директоријумом **развој**, а право читања над осталим директоријумима.
- Корисник **милица** ради у рачуноводству и има пуне привилегије само над директоријумом **рачуноводство**, а право читања над осталим директоријумима.
- Корисник **стефан** ради као асистент и има право читања над свим директоријумима.

решење \_\_\_\_\_

```

1      # groupadd продаја
2      # groupadd маркетинг
3      # groupadd развој
4      # groupadd рачуноводство
5      # groupadd излети
6
7      # useradd марко
8      # usermod -a -G продаја,излети марко
9      # passwd марко
10
11     # useradd ана
12     # usermod -a -G маркетинг,излети ана
13     # passwd ана
14
15     # useradd петар
16     # usermod -a -G развој,излети петар
17     # passwd петар
18
19     # useradd милица
20     # usermod -a -G рачуноводство,излети милица
21     # passwd милица
22
23     # useradd стефан
24     # usermod -a -G излети стефан
25     # passwd стефан
26
27     # chown директор:продаја /компанија/продаја
28     # chown директор:маркетинг /компанија/маркетинг
29     # chown директор:развој /компанија/развој
30     # chown директор:рачуноводство /компанија/рачуноводство
31     # chown директор:излети /компанија/излети
32
33     # chmod 777 /компанија/излети
34     # chmod 754 /компанија/продаја
35     # chmod 754 /компанија/маркетинг
36     # chmod 754 /компанија/развој
37     # chmod 754 /компанија/рачуноводство

```

решење \_\_\_\_\_

**Задатак 4** – [по 2% за сваки тачан од десет одговора, односно 20% укупно]

[2%] — Приказати све тагове у репозиторијуму.

решење \_\_\_\_\_

```

1      $ git tag

```

решење \_\_\_\_\_

[2%] — Обрисати грану стара\_функција.

решење \_\_\_\_\_

```

1      $ git branch -d стара_функција

```

решење \_\_\_\_\_

[2%] — Приказати информације о последњем commit-у у скраћеном формату.

решење \_\_\_\_\_

```

1      $ git log -1 --oneline

```

решење \_\_\_\_\_

[2%] — Додати све измењене датотеке у индекс.

решење \_\_\_\_\_

```

1      $ git add -u

```

решење \_\_\_\_\_

[2%] — Приказати разлике између индекса и последњег commit-a.

1	<code>\$ git diff --cached</code>	решење
---	-----------------------------------	--------

[2%] — Направи commit који исправља грешку из претходног commit-a. Претпоставити да је грешка већ исправљена, и да су измене додате у индекс. Нека нови commit буде обележен истом поруком као и оригинални.

1	<code>\$ git commit --amend --no-edit</code>	решење
---	--	--------

[2%] — Приказати списак датотека промењених у последњем commit-у.

1	<code>\$ git log --name-only</code>	решење
---	-------------------------------------	--------

[2%] — Излистати све удаљене репозиторијуме повезане са локалним репозиторијумом.

1	<code>\$ git remote -v</code>	решење
---	-------------------------------	--------

[2%] — Приказати историју commit-ова у оквиру којих је промењена датотека config.py.

1	<code>\$ git log -- config.py</code>	решење
---	--------------------------------------	--------

[2%] — Приказати све commit-ове између два тага v1.0 и v2.0.

1	<code>\$ git log v1.0..v2.0</code>	решење
---	------------------------------------	--------