

PROGRAMIRANJE

ZBIRKA ZADATAKA

Ishod učenja 1

2023-2024.

Zadatak 1

Napišite program koji od korisnika učitava podatke o dva stošca: polumjer baze (r) i visinu (h). Neka program izračuna volumene obaju stožaca koristeći formulu: $V = \frac{1}{3}\pi r^2 h$. Ispišite oba volumena i podatak o tome koji stožac ima veći volumen. Primjer rada programa:

```
Upisite polumjer baze prvog stosca: 0.9
Upisite visinu prvog stosca: 5.5
Upisite polumjer baze drugog stosca: 5.5
Upisite visinu drugog stosca: 0.9
Prvi stozac ima volumen 4.665265090580843
Drugi stozac ima volumen 28.50995333132737
Drugi stozac ima veci volumen
```

Moguće rješenje:

```
import math

r1 = float(input("Upisite polumjer baze prvog stosca: "))
h1 = float(input("Upisite visinu prvog stosca: "))
r2 = float(input("Upisite polumjer baze drugog stosca: "))
h2 = float(input("Upisite visinu drugog stosca: "))

v1 = (1 / 3) * math.pi * math.pow(r1, 2) * h1
v2 = (1 / 3) * math.pi * math.pow(r2, 2) * h2

print("Prvi stozac ima volumen", v1)
print("Drugi stozac ima volumen", v2)

if v1 > v2:
    print("Prvi stozac ima veci volumen")
elif v1 < v2 :
    print("Drugi stozac ima veci volumen")
else:
    print("Stosci imaju jednake volumene")
```

Zadatak 2

Napišite program koji od korisnika učitava x i y koordinate neke točke. Pretpostavite da postoji pravokutnik čije su x i y koordinate lijevog gornjeg kuta jednak 10, 10 te čija je širina 8 i visina 4. Izračunajte i ispišite nalazi li se točka u koliziji s pravokutnikom ili ne. Točka se nalazi u koliziji s pravokutnikom ako su ispunjeni svi sljedeći uvjeti:

- x pozicija točke je desno od lijevog ruba pravokutnika.
- x pozicija točke je lijevo od desnog ruba pravokutnika.
- y pozicija točke je ispod gornjeg ruba pravokutnika.
- y pozicija točke je iznad donjeg ruba pravokutnika.

Pretpostavite da je ishodište 0, 0 u gornjem lijevom kutu.

Moguće rješenje:

```
x = float(input("Upisite x: "))
y = float(input("Upisite y: "))

px = 10.0
py = 10.0
pw = 8.0
ph = 4.0

if x >= px and x <= px + pw and y >= py and y <= py + ph:
    print("Postoji kolizija")
else:
    print("Nema kolizije")
```

Zadatak 3

Napišite program koji od korisnika učitava broj kartice i PIN. Ispišite mu stanje računa, pitajte ga koliko novca želi podići, ispišite da ste mu isplatili traženi novac (ili da nema dovoljno) te mu na kraju ispišite preostalo stanje računa. Ispravni brojevi kartica i PIN-ovi su:

- Ako je broj kartice 5399-8500-1722-0325 i PIN 1234, onda korisnik ima na računu 350.50 eura,
- Ako je broj kartice 5399-1159-2036-1151 i PIN 0752, onda korisnik ima na računu 9510.00 eura.

Ako korisnik unese neispravan broj kartice ili PIN, ispišite da ste mu uzeli karticu, da se javi u poslovnici banke i završite program. Primjer rada programa kad korisnik ima dovoljno sredstava:

Upisite broj kartice: 5399-8500-1722-0325

Upisite PIN: 1234

Vase trenutno stanje je: 350.5 EUR

Koliko novca zelite podici: 200

Isplaceno vam je 200 EUR

Vase trenutno stanje je: 150.5 EUR

Primjer rada programa kad korisnik nema dovoljno sredstava:

Upisite broj kartice: 5399-1159-2036-1151

Upisite PIN: 0752

Vase trenutno stanje je: 9510 EUR

Koliko novca zelite podici: 10000

Nema dovoljno sredstava na racunu za trazenu isplatu

Vase trenutno stanje je: 9510 EUR

Primjer rada programa kad korisnik upiše krive podatke:

Upisite broj kartice: 5399-1159-2036-1151

Upisite PIN: 9999

Neispravan broj kartice ili PIN, vasa kartica je zadržana. Molimo, javite se u poslovnici banke.

Moguće rješenje:

```
broj_kartice = input("Upisite broj kartice: ")
pin = input("Upisite PIN: ")

stanje = 0.0
ok = True

if broj_kartice == "5399-8500-1722-0325" and pin == "1234":
    stanje = 350.50
elif broj_kartice == "5399-1159-2036-1151" and pin == "0752":
    stanje = 9510.00
else:
    print("Neispravan broj kartice ili PIN, vasa kartica je zadržana. Molimo,
javite se u poslovnici banke.")
    ok = False

if ok:
    print("Vase trenutno stanje je:", stanje, "EUR")

    isplata = float(input("Koliko novca zelite podici: "))
```

```
if isplata <= stanje:
    stanje -= isplata
    print("Isplaceno vam je", isplata, "EUR")
else:
    print("Nema dovoljno sredstava na racunu za trazenu isplatu")

print("Vase trenutno stanje je:", stanje, "EUR")
```

Zadatak 4

Napišite igru u kojoj je korisnik udaljen 20 koraka od cilja. U svakoj rundi omogućite korisniku da baci kockicu na kojoj može dobiti broj između 1 i 6 te ga pomaknite toliko polja prema cilju. Korisnik može doći do cilja samo ako dobije točan broj (primjerice, ako je udaljen 4 polja od cilja, mora dobiti 4 da dođe do cilja). Nacrtajte svaku rundu. Primjer odigravanja partije:

Runda: 1

Dobili ste 2

-x-----

Pritisnite enter za sljedeću rundu

Runda: 2

Dobili ste 5

-----x-----

Pritisnite enter za sljedeću rundu

Runda: 3

Dobili ste 4

-----x-----

Pritisnite enter za sljedeću rundu

Runda: 4

Dobili ste 1

-----x-----

Pritisnite enter za sljedeću rundu

Runda: 5

Dobili ste 3

-----x-----

Pritisnite enter za sljedeću rundu

Runda: 6

Dobili ste 1

-----x-----

Pritisnite enter za sljedeću rundu

Runda: 7

Dobili ste 1

-----x-----

Pritisnite enter za sljedeću rundu

Runda: 8

Dobili ste 2

-----x-----

Pritisnite enter za sljedeću rundu

Runda: 9

Dobili ste 5

Dobili ste prevelik broj, morate čekati sljedeću rundu

-----x-----

Pritisnite enter za sljedeću rundu

Runda: 10

```

Dobili ste 6
Dobili ste prevelik broj, morate cekati sljedecu rundu
-----x-
Pritisnite enter za sljedecu rundu

Runda: 11
Dobili ste 4
Dobili ste prevelik broj, morate cekati sljedecu rundu
-----x-
Pritisnite enter za sljedecu rundu

Runda: 12
Dobili ste 6
Dobili ste prevelik broj, morate cekati sljedecu rundu
-----x-
Pritisnite enter za sljedecu rundu

Runda: 13
Dobili ste 1
Pobjeda!

```

Moguće rješenje:

```

import random

udaljenost_cilja = 50
polozaj_igraca = 0
runda = 1
pobjeda = False

while True:
    print("Runda:", runda)

    dobiven_broj = random.randint(1, 6) # Baci kockicu
    print("Dobili ste", dobiven_broj)

    if polozaj_igraca + dobiven_broj < udaljenost_cilja:
        polozaj_igraca += dobiven_broj
    elif polozaj_igraca + dobiven_broj == udaljenost_cilja :
        print("Pobjeda!")
        break
    else:
        print("Dobili ste prevelik broj, morate cekati sljedecu rundu")

    # Crtanje
    for i in range(polozaj_igraca - 1) : # Predjeni put
        print("-", end = "")

    if polozaj_igraca > 0:
        print("x", end = "") # Polozaj igraca

    for i in range(udaljenost_cilja - polozaj_igraca) : # Preostali put
        print("-", end = "")

    print()
    input("Pritisnite enter za sljedecu rundu")
    runda += 1

```

Zadatak 5

Napišite program koji korisniku omogućuje rad s tekućim računom. Omogućite korisniku sljedeće akcije, dok god to želi:

- Ispis stanja računa.
- Dodavanje novca na račun.
- Isplatu novca s računa.
- Završetak programa.

Primjer rada programa:

A = ispis stanja

B = dodavanje novca na račun

C = isplata novca s računa

D = kraj

> A

Stanje racuna: 0 EUR

> B

Unesite iznos koji zelite dodati: 200

Iznos uspjesno dodan

> B

Unesite iznos koji zelite dodati: 100

Iznos uspjesno dodan

> C

Unesite iznos koji zelite isplatiti: 500

Nedovoljno sredstava

> A

Stanje racuna: 300 EUR

> C

Unesite iznos koji zelite isplatiti: 150

Iznos uspjesno isplacen

> C

Unesite iznos koji zelite isplatiti: 21

Iznos uspjesno isplacen

> A

Stanje racuna: 129 EUR

> D

Moguće rješenje:

```
tekuci_racun = 0.0
```

```
print("A = ispis stanja")
print("B = dodavanje novca na racun")
print("C = isplata novca s racuna")
print("D = kraj")
```

```
while True:
    operacija = input("\n> ")

    if operacija == "A":
        print("Stanje racuna:", tekuci_racun, "EUR")
    elif operacija == "B":
        iznos = float(input("Unesite iznos koji zelite dodati: "))
        tekuci_racun += iznos
        print("Iznos uspjesno dodan")
    elif operacija == "C":
        iznos = float(input("Unesite iznos koji zelite isplatiti: "))
        if tekuci_racun < iznos:
            print("Nedovoljno sredstava")
        else:
            tekuci_racun -= iznos
            print("Iznos uspjesno isplacen")
    elif operacija == "D":
        break
    else:
        print("Neispravna operacija")
```

Zadatak 6

Napišite program koji simulira aparat za parking. Omogućite korisniku da upiše koliko novca je ubacio te da upiše broj sati parkiranja (jedan sat je 2 EUR). Ispišite korisniku sve novčanice i kovanice koje će mu aparat vratiti, uz pretpostavku da će uvijek vratiti najmanji mogući broj (primjerice, ako treba vratiti 8 EUR onda će vratiti novčanicu od 5 EUR te kovanice od 2 i 1 EUR). Pretpostavite da aparat vraća samo novčanice od 10 EUR i 5 EUR te kovanice od 2 EUR i 1 EUR. Napomena: neka vam tipovi podataka budu int. Tko želi, može ih postaviti na double pa vidjeti s kojim problemom se tada suočavamo. Primjer rada programa:

```
Ubacite novac: 50
Koliko sati ste ostavili auto na parking: 6
Trebam vratiti: 38
Vracam novčanicu od 10 EUR
Vracam novčanicu od 10 EUR
Vracam novčanicu od 10 EUR
Vracam novčanicu od 5 EUR
Vracam kovanicu od 2 EUR
Vracam kovanicu od 1 EUR
```

Moguće rješenje:

```
ubaceno = int(input("Ubacite novac: "))
sati = int(input("Koliko sati ste ostavili auto na parking: "))
ok = True

cijena = sati * 2

if ubaceno < cijena:
    print("Neispravan odabir")
    ok = False

if ok:
    za_vratiti = ubaceno - cijena
    print("Trebam vratiti:", za_vratiti)

    while za_vratiti > 0:
        if za_vratiti > 10:
            print("Vracam novčanicu od 10 EUR")
            za_vratiti -= 10
        elif za_vratiti > 5:
            print("Vracam novčanicu od 5 EUR")
            za_vratiti -= 5
        elif za_vratiti > 2:
            print("Vracam kovanicu od 2 EUR")
            za_vratiti -= 2
        else:
            print("Vracam kovanicu od 1 EUR")
            za_vratiti -= 1
```

Zadatak 7

Napišite program koji zamisli jedan cijeli broj između 1 i 100. Neka program od korisnika u petlji učitava cijeli broj i ispisuje korisniku je li zamišljeni broj veći, manji ili jednak učitanoj broju. Petlja treba trajati sve dok korisnik ne pogodi zamišljeni broj. Na kraju, program treba ispisati u koliko pokušaja je korisnik pogodio broj. Primjer rada programa:

```
Pokusajte pogoditi broj: 50
Zamisljeni broj je veci
Pokusajte pogoditi broj: 75
Zamisljeni broj je veci
Pokusajte pogoditi broj: 88
Zamisljeni broj je manji
Pokusajte pogoditi broj: 81
Zamisljeni broj je veci
Pokusajte pogoditi broj: 85
Zamisljeni broj je manji
Pokusajte pogoditi broj: 83
Zamisljeni broj je veci
Pokusajte pogoditi broj: 84
Pogodili ste zamisljeni broj iz 7 pokusaja
```

Moguće rješenje:

```
import random

zamisljen = random.randint(1, 100)
broj_pokusaja = 0
pogodjeno = False

while pogodjeno == False:
    pokusaj = int(input("Pokusajte pogoditi broj: "))
    broj_pokusaja += 1

    if pokusaj > zamisljen:
        print("Zamisljeni broj je manji")
    elif pokusaj < zamisljen:
        print("Zamisljeni broj je veci")
    else:
        print("Pogodili ste broj iz", broj_pokusaja, "pokusaja")
        pogodjeno = True
```

Zadatak 8

Napišite program koji od korisnika učitava jedan string Z i jedan broj B. Neka program ispiše string Z točno B puta tako da između svaka dva ispisa stringa stavi crticu, osim iza zadnjeg. Primjerice, ako je Z = "MMM" i B = 3, onda treba ispisati: MMM-MMM-MMM

Moguće rješenje:

```
z = input("Upisite neki string: ")
b = int(input("Upisite neki prirodni broj: "))

for i in range(b):
    print(z, end="")

    if i < b - 1:
        print("-", end="")
```

Zadatak 9

Napišite program koji od korisnika učitava broj N veći od 1. Neka program zvjezdicama nacrtava prazni kvadrat sa stranicama duljine N. Primjerice, ako korisnik upiše N jednak 6, treba nacrtati ovakav kvadrat:

```
*****
*      *
*      *
*      *
*      *
*****
```

Moguće rješenje:

```
n = int(input("Upisite n veci od 1: "))

for redak in range(1, n + 1):
    # Svaki redak zapocinje zvjezdicom
    print("*", end = "")

    # Prvi i zadnji redak imaju n - 2 zvjezdice, a svi ostali reci imaju razmake
    for i in range(n - 2):
        if redak == 1 or redak == n:
            print("*", end = "")
        else:
            print(" ", end = "")

    # Svaki redak završava zvjezdicom
    print("*")
```

Zadatak 10

Napišite program koji od korisnika učitava neparni broj N veći ili jednak 3. Neka program zvjezdicama nacrtaj prazni kvadrat sa stranicama duljine N, okomito podijeljen na dva jednaka dijela znakom |.

Primjer za N=3	Primjer za N=5	Primjer za N=7
*** * * ***	***** * * * * * * *****	***** * * * * * * * * * * *****

Moguće rješenje:

```
n = int(input("Upisite neparni n veci ili jednak 3: "))

for redak in range(1, n + 1):
    print("*", end = "")

    # Prvi i zadnji redak imaju (n - 2) / 2 zvjezdica, a ostali imaju razmake
    for i in range(int((n - 2) / 2)):
        if redak == 1 or redak == n:
            print("*", end = "")
        else:
            print(" ", end = "")

    # Prvi i zadnji redak imaju 1 zvjezdicu, svi ostali "|".
    if redak == 1 or redak == n:
        print("*", end = "")
    else:
        print("|", end = "")

    # Ponovimo zvjezdice ili razmake, kao i za prvu polovicu
    for i in range(int((n - 2) / 2)):
        if redak == 1 or redak == n:
            print("*", end = "")
        else:
            print(" ", end = "")

    print("*") # Svaki redak završava zvjezdicom
```

Zadatak 11

Napišite program koji od korisnika učitava neparni broj N veći ili jednak 3. Neka program zvjezdicama nacrtá krunu sastavljenu od dva pravokutna trokuta s obje katete jednake N , okrenuta hipotenuzama jedan prema drugome. Upute:

- Prvi redak je iznimka jer ima zvjezdice na početku i kraju.
- Zadnji redak je iznimka jer ima sve zvjezdice.
- Svi ostali redci imaju po četiri zvjezdice i razmake.

Primjer za N=3	Primjer za N=5	Primjer za N=7
* * ** ** *****	* * ** ** * * * * * * * * *****	* * ** ** * * * * * * * * * * * * * * * * *****

Moguće rješenje:

```
n = int(input("Upisite neparni n veci ili jednak 3: "))

razmaka_izmedju = 2 * n - 4 # Racunamo od drugog retka
razmaka_unutar = 0

for redak in range(1, n + 1):
    if redak == 1: # Prvi redak je iznimka
        print("*", end="")
        for i in range(2 * n - 2):
            print(" ", end="")
        print("*", end="")
    elif redak == n: # Zadnji redak je iznimka
        for i in range(2 * n):
            print("*", end="")
    else: # Svi ostali redci imaju jednak oblik
        print("*", end="") # Prvo zvjezdica

        for i in range(razmaka_unutar): # Unutarnji razmaci
            print(" ", end="")

        print("*", end="") # Opet zvjezdica

        for i in range(razmaka_izmedju): # Vanjski razmaci
            print(" ", end="")

        print("*", end="") # Opet zvjezdica

        for i in range(razmaka_unutar): # Unutarnji razmaci
            print(" ", end="")

        print("*", end="") # Kraj

        razmaka_unutar += 1
        razmaka_izmedju -= 2
    print()
```

Zadatak 12

Napišite program koji od korisnika učitava broj N i zvjezdicama nacrtava dijamant visine $2*N - 1$. Prvi i zadnji redi su iznimke jer imaju samo jednu zvjezdicu. Svi ostali redi se crtaju po istom principu.

Primjer za N=3	Primjer za N=4	Primjer za N=5
<pre>* * * * * * * *</pre>	<pre> * * * * * * * * * * * *</pre>	<pre> * * * * * * * * * * * * * * * *</pre>

Moguće rješenje:

```
n = int(input("Upisite n: "))

razmaka_ispred = n - 1
razmaka_unutar = -1

for redak in range(1, 2 * n):

    # Uvijek zapocinjemo s razmacima
    for i in range(razmaka_ispred):
        print(" ", end="")

    # Uvijek ide prva zvjezdica
    print("*", end="")

    if redak > 1 and redak < 2 * n - 1: # Prvi i zadnji red su iznimka

        for i in range(razmaka_unutar):
            print(" ", end="")

        print("*", end="")

    print()

    if redak < n:
        razmaka_ispred -= 1
        razmaka_unutar += 2
    else:
        razmaka_ispred += 1
        razmaka_unutar -= 2
```


Zadatak 13

U prirodi, s obzirom na DNA sekvencu, RNA polimeraza prepisuje sve nukleotide (T, C, G i A) u novu RNA sekvencu. Napišite program koji simulira ovaj proces prepisivanja za DNA sekvencu koju unese korisnik. Neka korisnik unosi znakove koji predstavljaju nukleotide, dok god to želi. Svaki put kada korisnik unese ispravni nukleotid, ispišite u konzolu prepisani nukleotid prema sljedećim pravilima:

- T se prepisuje u U
- G se prepisuje u G
- C se prepisuje u C
- A se prepisuje u A

Ako korisnik unese neispravan nukleotid (vrijednost koja nije T, C, G ili A), ispišite odgovarajuću poruku o pogrešci i dopustite korisniku da unese novi nukleotid.

Moguće rješenje:

```
while True:
    nukleotid = input("Unesite nukleotid (T, G, C ili A) ili X za izlaz: ")

    if nukleotid == "T":
        print("Prepisan u: U")
    elif nukleotid == "G":
        print("Prepisan u: G")
    elif nukleotid == "C":
        print("Prepisan u: C")
    elif nukleotid == "A":
        print("Prepisan u: A")
    elif nukleotid == "X":
        break
    else:
        print("Neispravan nukleotid")
```

Zadatak 14

Neka korisnik prvo unese širinu i visinu ploče, a zatim i x i y koordinate jedne točke. Nacrtajte ploču zadane širine i visine. Koristite znak 'x' za ispis zadane točke te '_' za ispis svih ostalih praznih ćelija. Pretpostavimo da prvi znak u prvom redu ima koordinate (0, 0).

Moguće rješenje:

```
sirina = int(input("Upisite sirinu ploce: "))
visina = int(input("Upisite visinu ploce: "))

x = int(input("Upisite x tocke: "))
y = int(input("Upisite y tocke: "))

for redak in range(visina):
    for stupac in range(sirina):
        if stupac == x and redak == y:
            print("x", end="")
        else:
            print("_", end="")

    print()
```

Zadatak 15

Omogućuje korisniku da unese krvnu grupu donora i primatelja (A, B, AB, 0) te odredite i ispišite može li donor dati krv primatelju. Ovo su pravila:

- Tip 0 može donirati bilo kojem tipu.
- Tip A može donirati A i AB.
- Tip B može donirati B i AB.
- Tip AB može donirati samo AB.

Moguće rješenje:

```
donor = input("Upisite donorovu krvnu grupu: ")
primatelj = input("Upisite primateljevu krvnu grupu: ")

if donor == "0":
    print("Moze donirati")
elif donor == "A":
    if primatelj == "A" or primatelj == "AB":
        print("Moze donirati")
    else:
        print("Ne moze donirati")
elif donor == "B":
    if primatelj == "B" or primatelj == "AB":
        print("Moze donirati")
    else:
        print("Ne moze donirati")
elif donor == "AB":
    if primatelj == "AB":
        print("Moze donirati")
    else:
        print("Ne moze donirati")
else:
    print("Neispravna krvna grupa")
```

Zadatak 16

U biologiji je moguće predvidjeti rast populacije neke vrste. Napišite program koji od korisnika traži unos početne veličine populacije (N_0), stope rasta (r) i proteklog vremena (t). Izračunajte broj jedinki nakon određenog vremena pomoću sljedeće formule:

$$N = N_0 * e^{rt}$$

Gdje je e približno jednako 2,71828, a t je izražen u godinama. Na primjer, u nekoj bari je u početku 85 žaba, tako da je $N_0=85$. Njihova populacija raste za 18% svake godine, tako da je $r=0,18$. To znači da će njihova populacija nakon tri godine ($t=3$) biti 145 žaba.

Moguće rješenje:

```
import math

no = int(input("Unesite početnu populaciju: "))
r = float(input("Unesite stopu rasta: "))
t = float(input("Unesite proteklo vrijeme u godinama: "))

finalna_populacija = no * math.pow(math.e, r * t)

print("Nakon", t, "godina, populacija ce biti:", int(finalna_populacija))
```

Zadatak 17

Napišite jednostavnu igru u kojoj korisnik može držati dva predmeta, po jedan u svakoj ruci. U beskonačnoj petlji ponudite sljedeće opcije korisniku:

- Uzmi predmet u lijevu ruku. Ako korisnik već ima predmet u toj ruci, pitajte ga želi li baciti postojeći predmet.
- Uzmi predmet u desnu ruku. Ako korisnik već ima predmet u toj ruci, pitajte ga želi li baciti postojeći predmet.
- Zamijeni predmete iz lijeve u desnu i iz desne u lijevu.

Nakon svake operacije, ispišite predmete iz obje ruke. Primjer mogućeg toka programa:

A = uzmi predmet u lijevu ruku

B = uzmi predmet u desnu ruku

C = zamijeni predmete

> A

Unesite predmet: copper sword

Lijeva ruka: copper sword

Desna ruka:

> A

Vec drzite predmet - zelite li ga odbaciti (da/ne)? da

Unesite predmet: iron sword

Lijeva ruka: iron sword

Desna ruka:

> A

Vec drzite predmet - zelite li ga odbaciti (da/ne)? ne

Lijeva ruka: iron sword

Desna ruka:

> B

Unesite predmet: wooden shield

Lijeva ruka: iron sword

Desna ruka: wooden shield

> C

Lijeva ruka: wooden shield

Desna ruka: iron sword

Moguće rješenje:

```
lijeva_ruka = ""
```

```
desna_ruka = ""
```

```
print("A = uzmi predmet u lijevu ruku")
```

```
print("B = uzmi predmet u desnu ruku")
```

```
print("C = zamijeni predmete")
```

```
while True:
```

```
    akcija = input("\n> ")
```

```
    if akcija == "A":
```

```
        if lijeva_ruka != "":
```

```
        baci = input("Vec drzite predmet - zelite li ga odbaciti (da/ne)? ")
        if baci == "da":
            lijeva_ruka = ""
    if lijeva_ruka == "":
        lijeva_ruka = input("Unesite predmet: ")

elif akcija == "B":
    if desna_ruka != "":
        baci = input("Vec drzite predmet - zelite li ga odbaciti (da/ne)? ")
        if baci == "da":
            desna_ruka = ""
    if desna_ruka == "":
        desna_ruka = input("Unesite predmet: ")

elif akcija == "C":
    temp = lijeva_ruka
    lijeva_ruka = desna_ruka
    desna_ruka = temp

print("Lijeva ruka: ", lijeva_ruka)
print("Desna ruka: ", desna_ruka)
```

Zadatak 18

Napišite program koji od korisnika traži unos broja generacija. Zatim izračunajte i ispišite koliko će se mutacija dogoditi u tom broju generacija, ako znamo da postoji 10% šanse da se mutacija dogodi u svakoj generaciji. Primjer mogućeg toka programa:

```
Koliko generacija: 1000  
Bilo je 94 mutacija u 1000 generacija.
```

Moguće rješenje:

```
import random  
  
generacija = int(input("Koliko generacija: "))  
mutacija = 0  
  
for i in range(generacija):  
  
    # Generiramo slučajni broj između 1 i 100  
    n = random.randint(1, 100)  
  
    # Provjerimo je li u 10%  
    if n > 90:  
        mutacija += 1  
  
print("Bilo je", mutacija, "mutacija u", generacija, "generacija.")
```

Zadatak 19

Pretpostavimo da kolegij ima četiri ishoda učenja, od kojih svaki ima 25 bodova dostupnih za prikupljanje. Tražite od korisnika da unese prikupljene bodove za svaki od ishoda učenja i zatim prikažite konačnu ocjenu korisnika za taj kolegij. Provjerite je li korisnik skupio najmanje 50% dostupnih bodova za svaki ishod učenja. Ocjene su sljedeće:

- Najmanje 50,01 za dovoljan (2)
- Najmanje 58,01 za dobar (3)
- Najmanje 75,01 za vrlo dobar (4)
- Najmanje 92,01 za izvrstan (5)

Primjer mogućeg toka programa:

```
Upisite bodove za ishod 1: 13
Upisite bodove za ishod 2: 21
Upisite bodove za ishod 3: 19.5
Upisite bodove za ishod 4: 22.75
Vrlo dobar (4)
```

Moguće rješenje:

```
i1 = float(input("Upisite bodove za ishod 1: "))
i2 = float(input("Upisite bodove za ishod 2: "))
i3 = float(input("Upisite bodove za ishod 3: "))
i4 = float(input("Upisite bodove za ishod 4: "))

min_prolaz = 25 / 2

if i1 < min_prolaz or i2 < min_prolaz or i3 < min_prolaz or i4 < min_prolaz:
    print("Pali ste")
else:
    total = i1 + i2 + i3 + i4

    if total >= 92.01:
        print("Izvrstan (5)")
    elif total >= 75.01:
        print("Vrlo dobar (4)")
    elif total >= 58.01:
        print("Dobar (3)")
    elif total >= 50.01:
        print("Dovoljan (2)")
    else:
        print("Pali ste")
```


Zadatak 20

Neka korisnik unese širinu i visinu ploče koja predstavljaju broj stupaca i redaka. Napišite program koji ispisuje karirani uzorak (izmjenjuju se simboli '#' i 'o') za ploču zadanih dimenzija. Primjer ispisa kada je širina=10 i visina=5:

```
#o#o#o#o#o  
o#o#o#o#o#  
#o#o#o#o#o  
o#o#o#o#o#  
#o#o#o#o#o
```

Moguće rješenje:

```
širina = int(input("Upisite širinu: "))  
visina = int(input("Upisite visinu: "))  
  
for redak in range(visina):  
    for stupac in range(širina):  
        if redak % 2 == 0:  
            if stupac % 2 == 0:  
                print("#", end="")  
            else:  
                print("o", end="")  
        else:  
            if stupac % 2 == 0:  
                print("o", end="")  
            else:  
                print("#", end="")  
  
    print()
```

Zadatak 21

U igri Dungeons and Dragons igrači koriste kockice s različitim stranama kako bi odredili ishod akcije (na primjer, ako kockica ima 20 strana, tada su mogući ishodi brojevi od 1 do 20). Napišite program koji simulira bacanje kockica tražeći od korisnika da unese broj strana za sve kockice i broj kockica. Zatim ispišite rezultate bacanja za svaku kockicu. Primjer:

```
Upisite broj strana: 20
Koliko kockica bacate: 3
Kockica 1 je pokazala 14
Kockica 2 je pokazala 7
Kockica 3 je pokazala 5
```

Moguće rješenje:

```
import random

strana = int(input("Upisite broj strana: "))
kockica = int(input("Koliko kockica bacate: "))

for k in range(1, kockica + 1):
    bacanje = random.randint(1, strana)
    print("Kockica", k, "je pokazala", bacanje)
```

Zadatak 22

Neka korisnik unese broj N koji određuje visinu trokuta. U svakom retku ispišite brojeve od 1 do broja retka, potencirane brojem retka. Na primjer:

- u prvom redu ispišite broj 1 na potenciju 1
- u drugom redu ispišite brojeve 1 2 na potenciju 2, dakle: 1 4
- u trećem redu ispišite 1 2 3 na potenciju 3, dakle: 1 8 27
- itd.

Primjerice, kada je N=5, treba ispisati:

```
1
1 4
1 8 27
1 16 81 256
1 32 243 1024 3125
```

Moguće rješenje:

```
import math

n = int(input("Upisite n: "))

for redak in range(1, n + 1):
    for i in range(1, redak + 1):
        print(int(math.pow(i, redak)), end=" ")

    print()
```



```
        print("-", end="")
    print()

# More
for redak in range(1, visina_more + 1):
    for stupac in range(1, sirina + 1):
        print("^", end="")
    print()
```

Zadatak 24

Napišite program koji ispisuje sve datume u 2023. godini u hrvatskom formatu: dan.mjesec.godina. Koristite sljedeće informacije:

- Siječanj, ožujak, svibanj, srpanj, kolovoz, listopad i prosinac imaju 31 dan.
- Travanj, lipanj, rujanj i studeni imaju 30 dana.
- Veljača ima 28 dana.

Ispis:

```
1.1.2023.  
2.1.2023.  
3.1.2023.
```

...

```
29.1.2023.  
30.1.2023.  
31.1.2023.
```

```
1.2.2023.  
2.2.2023.  
3.2.2023.
```

...

```
31.12.2023.
```

Moguće rješenje:

```
max_dan = 0
```

```
for mjesec in range(1, 13):
```

```
    if mjesec == 2:
```

```
        max_dan = 28
```

```
    elif mjesec == 1 or mjesec == 3 or mjesec == 5 or mjesec == 7 or mjesec == 8 or  
mjesec == 10 or mjesec == 12:
```

```
        max_dan = 31
```

```
    else:
```

```
        max_dan = 30
```

```
    for dan in range(1, max_dan + 1):
```

```
        print(str(dan) + "." + str(mjesec) + ".2023.")
```

```
print()
```