



19. siječnja 2016.

Školsko natjecanje / Osnovna škola (5. i 6. i 7. i 8. razred)

Primjena algoritama OŠ

OPISI ALGORITAMA



Agencija za odgoj i obrazovanje
Education and Teacher Training Agency



HRVATSKI SAVEZ
INFORMATIČARA



Ministarstvo znanosti,
obrazovanja i sporta



HRVATSKA
ZAJEDNICA
TEHNIČKE
KULTURE



5.1. Zadatak: Napoleon

Autor: Nikola Dmitrović

Lako se uoči da je rješenje zadatka razlika zadane godine (**G**) i proslavljenog rođendana (**R**).

Programski kod (pisan u Python 3.x)

```
R = int(input())
G = int(input())
print(G - R)
```

Potrebno znanje: naredba učitavanja, naredba ispisa, operator oduzimanja

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka ¹
192	164	87,29 %

5.2. Zadatak: Centum

Autor: Nikola Dmitrović

Zadatak možemo riješiti tako što ćemo za zadanu godinu provjeriti kojem stoljeću ona pripada. Znači, ako je godina između 1 i 100 tada je riječ o 1. stoljeću, ako je godina između 101 i 200 tada je riječ o 2. stoljeću te tako sve do zadnjeg slučaja kada se treba provjeriti je li godina između 2301 i 2305 za 24. stoljeće.

Programski kod (pisan u Pythonu 3.x)

```
G = int(input())
if G >= 1 and G <= 100: print(1)
if G >= 101 and G <= 200: print(2)
if G >= 201 and G <= 300: print(3)
if G >= 301 and G <= 400: print(4)
if G >= 401 and G <= 500: print(5)
if G >= 501 and G <= 600: print(6)
if G >= 601 and G <= 700: print(7)
if G >= 701 and G <= 800: print(8)
if G >= 801 and G <= 900: print(9)
if G >= 901 and G <= 1000: print(10)
if G >= 1001 and G <= 1100: print(11)
if G >= 1101 and G <= 1200: print(12)
if G >= 1201 and G <= 1300: print(13)
```

¹ ukupan broj osvojenih bodova svih natjecatelja podijeljen s ukupnim brojem bodova koji su se mogli osvojiti



```
if G >= 1301 and G <= 1400: print(14)
if G >= 1401 and G <= 1500: print(15)
if G >= 1501 and G <= 1600: print(16)
if G >= 1601 and G <= 1700: print(17)
if G >= 1701 and G <= 1800: print(18)
if G >= 1801 and G <= 1900: print(19)
if G >= 1901 and G <= 2000: print(20)
if G >= 2001 and G <= 2100: print(21)
if G >= 2101 and G <= 2200: print(22)
if G >= 2201 and G <= 2300: print(23)
if G >= 2301 and G <= 2305: print(24)
```

Međutim, zadatak se može riješiti na puno elegantniji način. Dovoljno je uočiti vezu između godine i stoljeća kojem pripada. Posebno treba paziti na godine za koje ta veza ne vrijedi (100, 200, ...).

Programski kod (pisan u Pythonu 3.x)

```
G = input()
if G % 100 == 0:
    print (G // 100)
else:
    print (G // 100 + 1)
```

Za kraj jedna zanimljivost. U zadatku je zadano ograničenje na godinu **G** oblika $1 \leq G \leq 2305$. Zašto baš 2305? Po čemu je ta godina važna? Tko će se roditi te godine? Ako otkriješ, javi se do 11. 2. 2016. sa svojeg skole.hr maila na mail ndmitrovic@infokup.hr.

Potrebno znanje: naredba odlučivanja

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
192	77	57,70 %

5.3. Zadatak: Pat

Autor: Nikola Dmitrović

Na početku, u osam varijabli (P1, P2, P3, P4, M1, M2, M3, M4) učitamo tražene podatke. Na osnovi tih podataka odredimo broj prijedjenih koraka za Pata i Mata na kraju prvog dana (P1, M1), na kraju prva dva dana (P1+P2, M1+M2), nakon prva tri dana (P1+P2+P3, M1+M2+M3) i na kraju perioda promatranja (P1+P2+P3+P4, M1+M2+M3+M4). Jednostavnim uspoređivanjem navedenih podataka dobivamo tražene odgovore.

Programski kod (pisan u Pythonu 3.x)



```
P1 = int(input())
P2 = int(input())
P3 = int(input())
P4 = int(input())

M1 = int(input())
M2 = int(input())
M3 = int(input())
M4 = int(input())

Pprvi = P1
Pdrugi = P1 + P2
Ptreci = P1 + P2 + P3
Pcetvrti = P1 + P2 + P3 + P4

Mprvi = M1
Mdrugi = M1 + M2
Mtreci = M1 + M2 + M3
Mcetvrti = M1 + M2 + M3 + M4

if Pprvi > Mprvi:
    print("PAT")
else:
    print("MAT")

if Pdrugi > Mdrugi:
    print("PAT")
else:
    print("MAT")

if Ptreci > Mtreci:
    print("PAT")
else:
    print("MAT")

if Pcetvrti > Mcetvrti:
    print("PAT")
else:
    print("MAT")
```

Potrebno znanje: naredba odlučivanja

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
192	84	60,58 %



6.1. Zadatak: Brojimo

Autor: Nikola Dmitrović

Sljedeći izgovoreni broj uvijek je za jedan veći od drugog učitano broj, osim u slučaju kada je taj broj 100 te brojanje kreće ispočetka, od broja jedan. Posebni slučaj bio je jedan od dummy test podataka (test podataka iz zadatka).

Programski kod (pisan u Pythonu 3.x)

```
X = int(input())
Y = int(input())
if Y == 100:
    print(1)
else:
    print(Y + 1)
```

Potrebno znanje: naredba odlučivanja

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
203	135	83,63 %

6.2. Zadatak: Pizza

Autor: Adrian Satja Kurdića

Jedno je rješenje s pomoću while petlje simulirati uzimanje komada pizze. Dok je broj komada pizze veći od nule, smanjujemo broj komada za jedan, a u varijabli *tko_je_na_redu* pratimo koja osoba uzima taj komad. Tu varijablu treba povećati za 1 (tj. pomaknuti na sljedeću osobu) pri svakom uzimanju komada, osim ako ona iznosi N, a u tom je slučaju treba ponovno postaviti na 1. Tu varijablu (*tko_je_na_redu*) ispisujemo prilikom uzimanja zadnjeg komada.

Drugo je rješenje matematičko: uz malo razmišljanja moguće je zaključiti da treba ispisati ostatak pri dijeljenju broja komada s brojem osoba, osim ako taj ostatak iznosi 0, a u tom slučaju valja ispisati N.

Programski kod (pisan u Pythonu 3.x) - 1. način

```
n = int(input())
k = int(input())
tko_je_na_redu = 1
while k > 0:
    if k == 1:
        print (tko_je_na_redu)
    k -= 1
    if tko_je_na_redu < n:
        tko_je_na_redu += 1
    else:
```



```
tko_je_na_redu = 1
```

Programski kod (pisan u Pythonu 3.x) – 2. način

```
n = int(input())
k = int(input())
if k % n == 0:
    print(n)
else:
    print(k % n)
```

Potrebno znanje: while petlja

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
203	46	43,94%

6.3. Zadatak: Razdijeli

Autor: Nikola Dmitrović

Zadatak možemo riješiti tako da generiramo sve mogućnosti za uređenu trojku brojeva (A, B, C) pri čemu su A, B i C brojevi od 1 do $N - 2$ (gornja granica je $N - 2$ zbog uvjeta da su A, B i C najmanje jedan). Za svaku od njih provjerimo odgovara li zadanim uvjetima. Drugi način je da generiramo samo one mogućnosti za koje već vrijedi uvjet $A \leq B \leq C$ i da za svaku od njih provjerimo dodatni uvjet $A + B + C = N$.

Oba rješenja oslanjaju se na trostruku for petlju i pametno postavljanje granica za kontrolne varijable. Moguća su rješenja s dvije for petlje, s drugačije postavljenim uvjetima na kontrolne varijable, ali njih ćemo prepustiti čitatelju.

Programski kod (pisan u Pythonu 3.x) – 1. način

```
N = int(input())
ukupno = 0
for i in range(1, N - 1):
    for j in range(1, N - 1):
        for k in range(1, N - 1):
            if i <= j <= k and i + j + k == N:
                ukupno += 1
print(ukupno)
```

Programski kod (pisan u Pythonu 3.x) – 2. način

```
N = int(input())
ukupno = 0
```



```
for i in range(1, N - 1):  
    for j in range(i, N - 1):  
        for k in range(j, N - 1):  
            if i + j + k == N:  
                ukupno += 1  
  
print(ukupno)
```

Potrebno znanje: višestruka naredba ponavljanja

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
203	26	14,51%

7.1. Zadatak: Centum

Vidi opis 5.2.

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
95	76	89,36%

7.2. Zadatak: Razdijeli

Vidi opis 6.3.

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
95	34	39,78%

7.3. Zadatak: Cesta

Autor: Adrian Satja Kurdija

Zadane udaljenosti susjednih gradova spremićemo u niz. Kada unesemo brojeve A i B, najprije ćemo ih međusobno zamijeniti ako je A veći od B, tako da A bude manji od B. Potom ćemo for petljom proći po nizu, i to po onim udaljenostima koje se nalaze između A i B, počevši od udaljenosti gradova A i A+1, pa do udaljenosti gradova B-1 i B, usput pribrajaćući te udaljenosti u varijablu *ukupno* koju na kraju ispisujemo.

Programski kod (pisan u Pythonu 3.x)

```
N = int(input())
```



```
cesta = list(map(int, input().split()))
A, B = map(int, input().split())
if A > B:
    A, B = B, A
    #print(A, B)
ukupno = 0
for i in range(A-1, B-1):
    ukupno += cesta[i]
print(ukupno)
```

Potrebno znanje: nizovi

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
95	35	43,04%

8.1. Zadatak: Stupac

Autor: Nikola Dmitrović

Zadatak možemo riješiti metodom iscrpljivanja tako da za svaki mogući ulaz odredimo izlaz. Pola bodova iz zadatka moguće je dobiti jednostavnom provjerom za svako slovo koji je stupac (A je prvi stupac, B je drugi..).

Za kraće opće rješenje treba uočiti da se nakon svakih 26 stupaca prvo slovo u oznaci promjeni u slovo koje slijedi iza njega u engleskoj abecedi. Na taj način možemo sve stupce podijeliti u pakete po 26 stupaca. Prvo slovo iz oznake određuje u kojem paketu od početka se nalazi zadani stupac, dok drugo slovo iz oznake označava koji je to stupac po redu u trenutnom paketu od 26 stupaca.

Programski kod (pisan u Pythonu 3.x)

```
N = int(input())
if N == 1:
    Z = input()
    print(ord(Z) - 64)
else:
    Z1 = input()
    Z2 = input()
    prvi = ord(Z1) - 64
    drugi = ord(Z2) - 64
    print(prvi * 26 + drugi)
```




Potrebno znanje: naredba odlučivanja, rad sa znakovima

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
97	48	58,55%

8.2. Zadatak: Cesta

Vidi opis 7.3.

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
97	46	54,66%

8.3. Zadatak: Baba

Autor: Adrian Satja Kurdiya

Prepravljena riječ počinjat će slovom A ili slovom B. Počinje li slovom A, sljedeća slova moraju biti B, A, B, A i tako dalje do kraja riječi. Počinje li slovom B, sljedeća slova moraju biti A, B, A, B i tako dalje do kraja riječi. Rješenje je isprobati obje navedene mogućnosti (riječ ABAB... i riječ BABA...): za svaku od njih, s pomoću for-petlje izbrojit ćemo broj promijenjenih slova (tj. pozicija na kojima se zadana riječ razlikuje od odabrane) i na kraju ispisati manji od tih dvaju brojeva.

Programski kod (pisan u Pythonu 3.x) – 1. prijedlog

```
s = input()
prva_mogucnost = druga_mogucnost = 0
for i in range(len(s)):
    if i % 2 == 0:
        prva_mogucnost += (s[i] != 'A')
        druga_mogucnost += (s[i] != 'B')
    else:
        prva_mogucnost += (s[i] != 'B')
        druga_mogucnost += (s[i] != 'A')
print(min(prva_mogucnost, druga_mogucnost))
```

Programski kod (pisan u Pythonu 3.x) – 2. prijedlog

```
rijec = input()
prva = "ABABABABABABABABABAB"
druga = "BABABABABABABABABABA"
```



```
ukupnoA = ukupnoB = 0
for i in range(len(rijec)):
    if rijec[i] != prva[i]:
        ukupnoA += 1
    if rijec[i] != druga[i]:
        ukupnoB += 1
print(min(ukupnoA, ukupnoB))
```

Potrebno znanje: stringovi

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
97	18	37,93%