19. siječnja 2017.

2017 natjecanje iz informatike

Državno

Školska razina / 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: Vezice — Autor: Nikola Dmitrović

Motivacijski zadatak za početak natjecanja. Razliku u duljinu između vezica odredit ćemo tako što ćemo od duljine duže vezice oduzeti duljinu kraće vezice. Naredbom odlučivanja ćemo odrediti koja je vezica (D ili L) dulja.

Programski kod (pisan u Python 3)

```
L = int(input())
D = int(input())
if L < D:
    razlika = D - L
else:
    razlika = L - D
print(razlika)
```

Potrebno znanje: naredba učitavanja i ispisivanja, apsolutna vrijednost ili naredba odlučivanja

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
199	**137**	**32.40**

5.2. Zadatak: Mogućnost — Autor: Adrian Satja Kurdija

Ukupan broj bodova dobiva se množenjem broja K s brojem test podataka koji su natjecatelju „prošli“. Dakle, rezultat X moguć je ako je on višekratnik broja K; drugim riječima, ako je njegov ostatak pri dijeljenju s K jednak 0. Broj X pritom mora biti manji ili jednak maksimalnom rezultatu koji iznosi N* K.

Programski kod (pisan u Pythonu 3)

```
N = int(input())
K = int(input())
X = int(input())
if X <= N * K and X % K == 0:
    print('DA')
else:
    print('NE')
```

Potrebno znanje: dijeljenje s ostatkom, naredba odlučivanja

Kategorija: ad hoc

Broj natjecatelja koji su rješavali	Broj natjecatelja koji su točno	Prosječna riješenost

zadatak	riješili zadatak	zadatka
199	**17**	**36.28**

5.3. Zadatak: Jules Autor: Nikola Dmitrović

Kada je broj A bliži broju C od broja B ako je npr. A ≤ C ≤ B? Jasno je da će A biti bliži C od B ako vrijedi: C - A < B - C. U ovom slučaju, lako ćemo analizom svih slučajeva zaključiti da su:

1. brojevi {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} bliži broju nula nego 25;
2. brojevi {13, 14, ..., 36, 37} bliži broju 25 nego nuli ili 50;
3. brojevi {38, 39, ..., 61, 62} bliži broju 50 nego 25 ili 75;
4. brojevi {63, 64, ..., 86, 87} bliži broju 75 nego 50 ili 100;
5. brojevi {88, 89, ..., 98, 99} bliži broju 100 nego 75.

Sada je potrebno samo dobro postaviti uvjete u naredbu odlučivanja. Ili provjeriti sve slučajeve ako ste raspoloženi za to.

Programski kod (pisan u Pythonu 3)

```
V = int(input())

if V <= 12:
    print("SKORO NISTA")

if V >= 13 and V <= 37:
    print("SKORO JEDNA CETVRTINA")

if V >= 38 and V <= 62:
    print("SKORO POLA")

if 63 <= V <= 87: #Python dozvoljava i ovakav zapis logičkog uvjeta
    print("SKORO TRI CETVRTINE")

if V >= 88:
    print("SKORO SVE")
```

Potrebno znanje: naredba odlučivanja

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
199	**36**	**42.74**

6.1. Zadatak: Funta — Autor: Nikola Dmitrović

Ako vrijedi da u jednoj funti ima **20** šilinga, a u jednom šilingu **12** penija, tada će se ukupan broj penija moći izračunati po formuli F * 20 * 12 + S * 12 + P.

Programski kod (pisan u Pythonu 3)

```
F = int(input())
S = int(input())
P = int(input())

print(F * 20 * 12 + S * 12 + P)
```

Potrebno znanje: naredba učitavanja, naredba ispisa, osnovne matematičke operacije

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
199	**153**	**31.73**

6.2. Zadatak: Haršadski — Autor: Nikola Dmitrović

Prirodan broj koji je **djeljiv zbrojem svojih znamenki** zovemo haršadskim brojem. Prvo ćemo odrediti zbroj znamenki zadanog broja te provjeriti vrijedi li zadani uvjet.

Programski kod (pisan u Pythonu 3)

```
N = int(input()
kopija = N              #nužno je sačuvati originalnu vrijednost broja N
s = 0
while kopija > 0:       #dok je broj kopija veći od nule
    s += kopija % 10    #zbroj znamenki povećamo za vrijednost znamenke jedinica
    kopija //= 10       #'smanjimo' broj kopija za skroz desnu znamenku
if N % s == 0:          #ako je broj N djeljiv sa zbrojem znamenki
    print(s)
else:
    print('ARTHUR')
```

Potrebno znanje: while naredba, algoritam određivanja sume znamenki

Kategorija: ad hoc

Za kraj jedna zanimljivost. U zadatku je potrebno ispisati poruku 'ARTHUR'. Zašto baš to ime? Ako znaš, javi se do 13. 3. 2017. sa svojeg skole.hr maila na mail ndmitrovic@infokup.hr.

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
199	**37**	**24.13**

6.3. Zadatak: Smotra — Autor: Adrian Satja Kurdija

U pomoćnim varijablama održavamo najbolji pronađeni par traženih brojeva: nazovimo ih *R_best* i *S_best*. Na početku možemo staviti *R_best = 1, S_best = N*. Potom *for* petljom na sve moguće načine odabiremo broj redova R. Za svaki taj odabir provjeravamo je li N djeljiv sa R (tj. je li ostatak pri njihovom dijeljenju 0): ako jest, tada je potencijalni broj stupaca S = N/R. Ako vrijedi i uvjet R ≤ S te ako je razlika S – R manja od dosad najmanje pronađene razlike *S_best - R_best*, spremit ćemo brojeve R i S kao dosad najbolje, dakle u varijable *R_best* i *S_best* koje na samom kraju ispisujemo.

Programski kod (pisan u Pythonu 3)

```
N = int(input())
R_best = 1
S_best = n
for r in range(2, N):
    if N % r == 0:
        s = N // r
        if r <= s and S_best - R_best > s - r:
            R_best = r
            S_best = s
print(R_best, S_best)
```

Potrebno znanje: iskušavanje svih mogućnosti, *for* petlja

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
199	**9**	**14.88**

7.1. Zadatak: Papiga — Autor: Adrian Satja Kurdija

Učitat ćemo string (niz znakova) te ga potom pet puta ispisati bez razmaka. Alternativno, možemo spajati string sa samim sobom koristeći operator + ako nam programski jezik to omogućuje.

Programski kod (pisan u Pythonu 3)

```
a = input()
print(a + a + a + a + a)
```

Potrebno znanje: unos i ispis stringa

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
132	**125**	**37.94**

7.2. Zadatak: 442 — Autor: Nikola Dmitrović

Promotrimo u zadanom nogometnom broju N znamenku jedinica, desetica i stotica. Ako je zbroj tih znamenki jednak 10, pronašli smo traženu trojku, ispišimo je redom kako je to zadano u zadatku i zaustavimo izvršenje programa. Ako zbroj nije deset, cjelobrojno podijelimo broj s deset i ponovimo opisani algoritam.

Kako je broj N nogometni sigurno ćemo pronaći takvu trojku. Ovaj algoritam zadovoljava i uvjet o traženju trojke koja se nalazi desno od svih drugih.

Programski kod (pisan u Pythonu 3)

```
N = int(input())
while N > 100:          # kako tražimo 'trojke' broj ne smije pasti ispod 100
    j = N % 10
    d = (N // 10) % 10
    s = (N // 100) % 10
    if j + d + s == 10:
        print(s, d, j)
        break
    N //= 10
```

Potrebno znanje: while naredba

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka

132	**44**	**30.29**

7.3. Zadatak: Bomboni — Autor: Adrian Satja Kurdija

Pronalazak najmanjeg elementa u nizu vršimo na standardan način: po nizu prolazimo *for* petljom te u pomoćnoj varijabli održavamo najmanji dotad viđeni element, kao i njegov indeks (poziciju), da bismo ga na kraju mogli udvostručiti. Prije udvostručenja treba još jednim prolaskom po nizu provjeriti je li pronađeni najmanji element jedinstven ili ih ima više. Ako ih ima više, nijedno dijete nije uplakano pa možemo ispisati konačni niz i završiti s računanjem.

Cijeli postupak treba, naravno, smjestiti unutar *while* petlje jer se može ponavljati. Petlju prekidamo (*break*) u gore navedenom slučaju (nema uplakanih). Da će se postupak ponavljati beskonačno možemo zaključiti na više načina, npr. ako se svaki element niza barem jednom udvostruči. Autor poziva mlade matematičare bilo kojeg uzrasta da pronađu (i dokažu) nužan i dovoljan uvjet koji kaže kakvi moraju biti početni brojevi da bi postupak trajao beskonačno.[1]

Programski kod (pisan u Pythonu 3)

```
N = int(input())
a = []
for i in range(N):
    a.append(int(input()))
broj_iteracija = 0
while a.count(min(a)) == 1:
    a[a.index(min(a))] *= 2
    broj_iteracija += 1
    if broj_iteracija > 10 * n:
        print('INFINITY')
        exit(0)
for i in range(n):
    print(a[i], end=' ')
print()
```

Potrebno znanje: nizovi, pronalazak najmanjeg elementa

Kategorija: simulacija

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
132	**13**	**18.74**

[1] Odgovore možete slati na askurdija@gmail.com.

8.1. Zadatak: Skoro — Autor: Nikola Dmitrović

Prvo odredimo vrijednost razlomka B/R te provjerimo kojoj je četvrtini bliža. Sve vrijednosti koje su manje od 0.125 ($(0+\frac{1}{4}):2=0.125$) su bliže nula četvrtina nego jednoj četvrtini. Ostale rubne vrijednosti su: $(\frac{1}{4}+\frac{2}{4}):2=0.375$, $(\frac{2}{4}+\frac{3}{4}):2=0.625$, $(\frac{3}{4}+\frac{4}{4}):2=0.875$.

Programski kod (pisan u Pythonu 3)

```
B, R = map(int, input().split())
razlomak = B / R
if razlomak < 0.125:
    print(0, 4)
elif razlomak < 0.375:
    print(1, 4)
elif razlomak < 0.625:
    print(2, 4)
elif razlomak < 0.875:
    print(3, 4)
else:
    print(4, 4)
```

Potrebno znanje: naredba odlučivanja

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
71	**26**	**27.66**

8.2. Zadatak: Sfenički — Autor: Nikola Dmitrović

Prirodan broj koji je **jednak umnošku triju različitih prostih brojeva** zovemo sfeničkim brojem. Za zadani sfenički broj trebalo je ispisati tri prosta broja koja su ga učinila takvim. Zadatak možemo riješiti tako da krenemo redom po svim prirodnom brojevima, tražimo one koje su prosti i kada nađemo neki takav podijelimo broj N s tim brojem. Postupak ponavljamo sve dok ne pronađemo tri prosta broja s kojima je N djeljiv, a možemo gledati i kada je broj N postao jedan.

Postoji i jednostavnije rješenje koje ne zahtjev traženje prostih brojeva. Prema definiciji, sfenički broj je takav da ga dijele samo tri prosta broja i zato nema potrebe za provjerom prostosti djelitelja.

Programski kod (pisan u Pythonu 3)

```
from math import *
def prost(N):
```

```
    for i in range(2, int(sqrt(N)) + 1):
        if N % i == 0:
            return 0
    return 1
N = int(input())
i = 2
while N != 1:
    if prost(i) and N % i == 0:
        N //= i
        print(i, end = ' ')
    i += 1
```

Potrebno znanje: while naredba

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
71	**33**	**36.38**

8.3. Zadatak: Školsko

Autor: Adrian Satja Kurdija

Na prvom zadatku učenik je osvojio najmanje 0, a najviše N1 test podataka. Na drugom zadatku učenik je osvojio najmanje 0, a najviše N2 test podataka. Na trećem zadatku učenik je osvojio najmanje 0, a najviše N3 test podataka. Rješenje je isprobati sve moguće varijante, tj. svaki mogući odabir triju brojeva od kojih je prvi između 0 i N1, drugi između 0 i N2, a treći između 0 i N3. To činimo trima ugnježđenim *for* petljama kao u priloženom kodu. Za svaku varijantu računamo broj osvojenih bodova, tj. ukupni rezultat.

Ako sve moguće ukupne rezultate spremimo u niz, potrebno je u tom nizu pronaći broj različitih elemenata. To možemo postići npr. sortiranjem niza po veličini, jer se tada „duplikati“ javljaju uzastopce pa ih je lako detektirati. Alternativno, možemo koristiti strukturu *set* koja automatski odbacuje duplikate.

Programski kod (pisan u Pythonu 3)

```
n1, k1 = map(int, input().split())
n2, k2 = map(int, input().split())
n3, k3 = map(int, input().split())
s = set()
for i in range(n1 + 1):
    for j in range(n2 + 1):
        for k in range(n3 + 1):
```

```
            s.add(i * k1 + j * k2 + k * k3)
print(len(s))
```

Potrebno znanje: iskušavanje svih mogućnosti, ugnježđena for petlja, sortiranje niza ili *set*

Kategorija: ad hoc

Broj natjecatelja koji su rješavali zadatak	Broj natjecatelja koji su točno riješili zadatak	Prosječna riješenost zadatka
71	**20**	**33.46**