

INFORMATIČKI KLUB  
**FUTURA**

```
#include<stdio.h> C  
int main()  
{ RADIONICE PROGRAMIRANJA  
  printf("Hello World!");  
  ZA SREDNJE ŠKOLE  
  return 0;  
}
```

# **RADIONICE PROGRAMIRANJA ZA SREDNJE ŠKOLE - 7. RADIONICA**

Nikola Rabrenović i Krunoslav Žubrinić  
Informatički klub FUTURA  
Dubrovnik, 7. veljače 2015.



# Creative Commons



## slobodno smijete:

- **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo
- **remiksirati** — prerađivati djelo



## pod slijedećim uvjetima:

- **imenovanje**. Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
- **nekomercijalno**. Ovo djelo ne smijete koristiti u komercijalne svrhe.
- **dijeli pod istim uvjetima**. Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.



U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu.

Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.

Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Tekst licence preuzet je s <http://creativecommons.org/>.

# Sadržaj

---

- Rješenje zadatka sa školskog natjecanja 2015.
  - Piton (1.-2. razred)
  
- Rješenje zadatka sa prethodnih županijskih natjecanja.
  - Tipke (1.-2. razred 2014.)
  - Kvadrat (1.-2. razred 2011.)

# Zadatak: Piton



- Piton je strukturirani programski jezik sintaksom sličan popularnom Pythonu koji ima samo dvije naredbe:
  - **Px** gdje je **x** malo slovo engleske abecede: ova naredba ispisuje slovo **x**.
  - **Fk** gdje je **k** prirodni broj manji ili jednak od 20, zapisan bez vodećih nula: ova naredba ponavlja točno **k** puta blok naredbi koji slijedi.
- Blokovi naredbi se specificiraju uvlačenjem naredbi, ali se koristi znak '.' (točka) umjesto uobičajenih razmaka.
  - Za nenegativni cijeli broj N definiramo N-uvučeni blok kao niz linija koji se sastoji od jedne ili više:
    - Naredbi P uvučene s N točaka.
    - Naredbi F uvučene s N točaka nakon koje slijedi točno jedan M-uvučeni blok gdje je M cijeli broj veći od N.
- Jedan blok može sadržavati proizvoljan broj P i F naredbi proizvoljnim redoslijedom.
- Nakon svake F naredbe slijedi blok koji mora biti više uvučen nego sama naredba.
- Prva linija ne sadrži točku).

# Zadatak: Piton



Napišite program koji će, za zadani program u Pitonu, pronaći **koliko kojih malih slova** zadani program ispisuje.

## ULAZNI PODACI:

- U prvom redu nalazi se broj ( $L \leq 100$ ) linija zadanog programa.
- Sljedećih L linija sadrži niz od najviše 80 znakova – programski kod programa.
  - Dozvoljeni znakovi su točke, mala slova engleske abecede, velika slova 'F' i 'P' te znamenke '0' do '9'.

## IZLAZNI PODACI:

- Potrebno je ispisati koliko se različitih malih slova pojavljuje u izlazu zadanog programa. Slova trebaju biti ispisana abecednim redosljedom.
  - Za svako slovo x koje se pojavljuje u izlazu potrebno je ispisati jedan red oblika '**x k**', gdje je k ukupan broj ispisivanja slova x u programu.



**Vrijeme**



# Zadatak: Piton



**ulaz**

5  
Px  
Py  
F3  
..Px  
..Pz

**izlaz**

x 4  
y 1  
z 3

**ulaz**

9  
F3  
..Pa  
..Pb  
..F4  
...Pc  
..F2  
.....Pd  
.....Pe  
..Pa

**izlaz**

a 6  
b 3  
c 12  
d 6  
e 6

**ulaz**

8  
Pa  
F3  
.F12  
..Px  
..F20  
...F2  
....F3  
.....Pa

**izlaz**

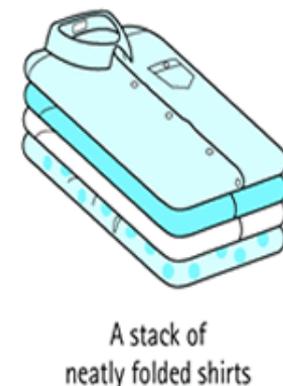
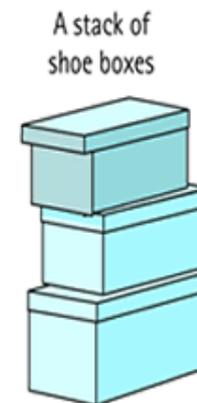
a 4321  
x 36

**Vrijeme** 



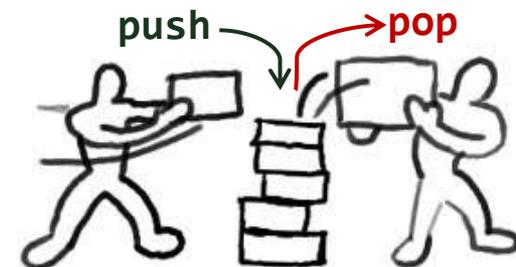
# Rješenje zadatka: Piton

- Problem provjere sintaksne ispravnosti računalnog programa.
- Za provjeru ispravnosti strukture računalnog programa uobičajeno se koristi struktura podataka **stog** (*stack*).



# Stog

- Struktura podataka kod koje se posljednji pohranjeni podatak prvi uzima u obradu
- Potrebne operacije:
  - dodavanje (**push**) elemenata na vrh stoga (*top*)
  - brisanje (**pop**) elemenata s vrha stoga
  - inicijalizacija praznog stoga
- Može se realizirati poljem
  - u jednodimenzionalno polje se dodaju ili brišu pojedine stavke prema načelu *Last In First Out (LIFO)*.
  - ažurira se vrijednost varijable koja predstavlja vrh stoga.
- Inicijalizacija praznog stoga:
  - postavljanje vrha na početnu vrijednost

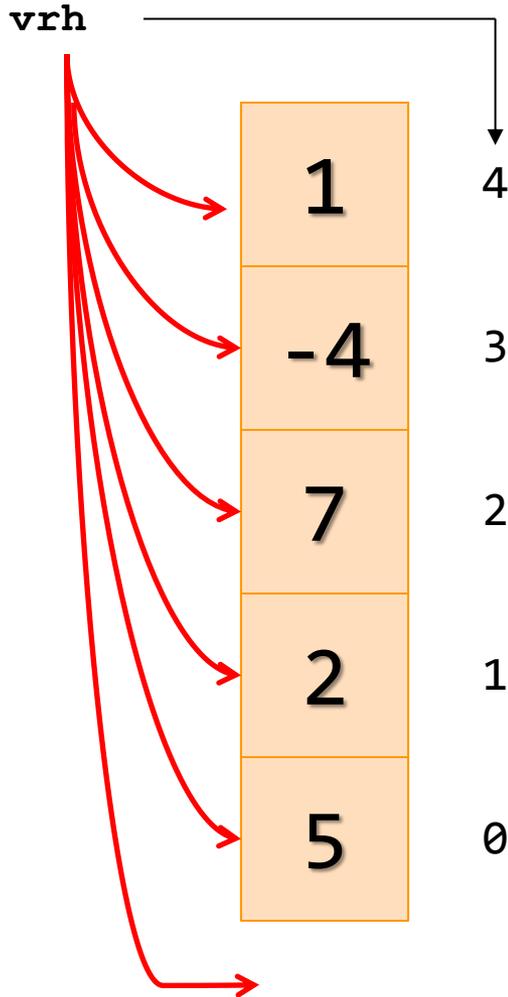


```
vrh = -1;
```

# Dodavanje elementa na stog



MAXSTOG=5



```
if (vrh < MAXSTOG-1) {  
    vrh++;  
    polje[vrh] = element;  
}
```

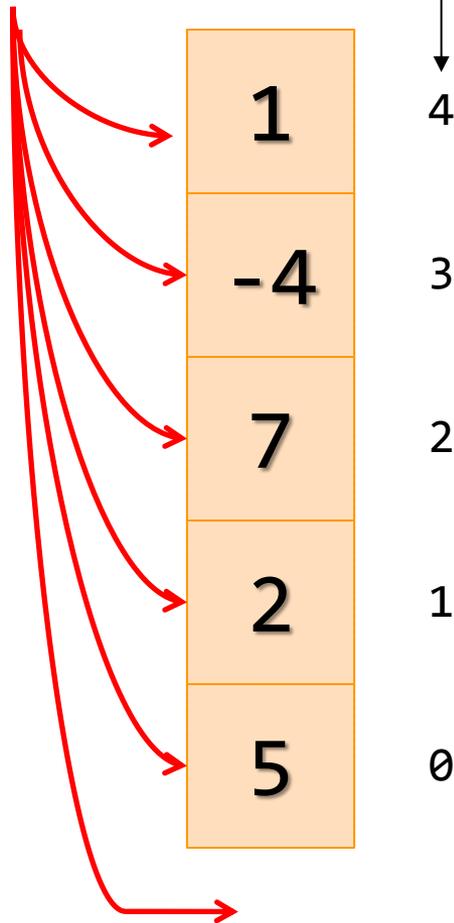
```
vrh = -1;  
dodaj(5);  
dodaj(2);  
dodaj(7);  
dodaj(-4);  
dodaj(1);  
dodaj(9);
```

# Skidanje elemenata sa stoga



MAXSTOG=5

vrh



```
if (vrh >= 0) {  
    podatak = polje[vrh];  
    vrh--;  
}
```

```
skini();  
skini();  
skini();  
skini();  
skini();  
skini();
```

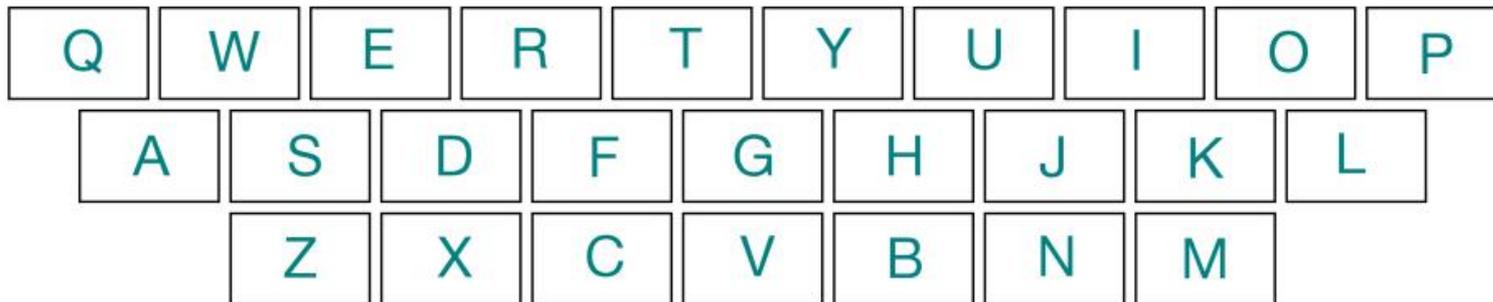
# Zadatak: Tipke



Pojavom tableta i mobitela s ekranima osjetljivima na dodir, fizičke tipkovnice lagano odlaze u zaborav. Međutim, jedan problem s tipkovnicama na ekranima je veća mogućnost pogreške prilikom tipkanja, posebno kada je riječ o relativno malenim ekranima.



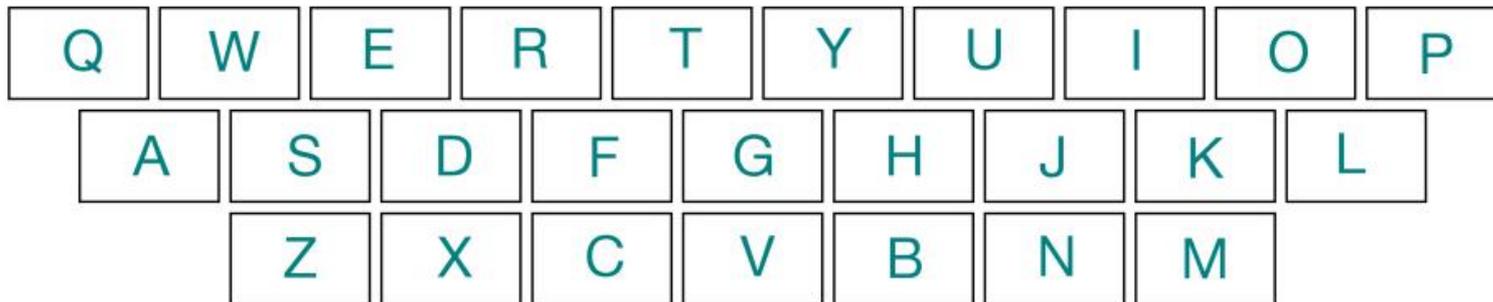
# Zadatak: Tipke



Za ovu priliku pretpostavimo da se tipkovnica sastoji od skupa pravokutnika u standardnom koordinatnom sustavu (koordinata  $X$  raste sa lijeva na desno, koordinata  $Y$  raste odozdo prema gore) kao na slici gore. Tipkovnica je definirana na sljedeći način:

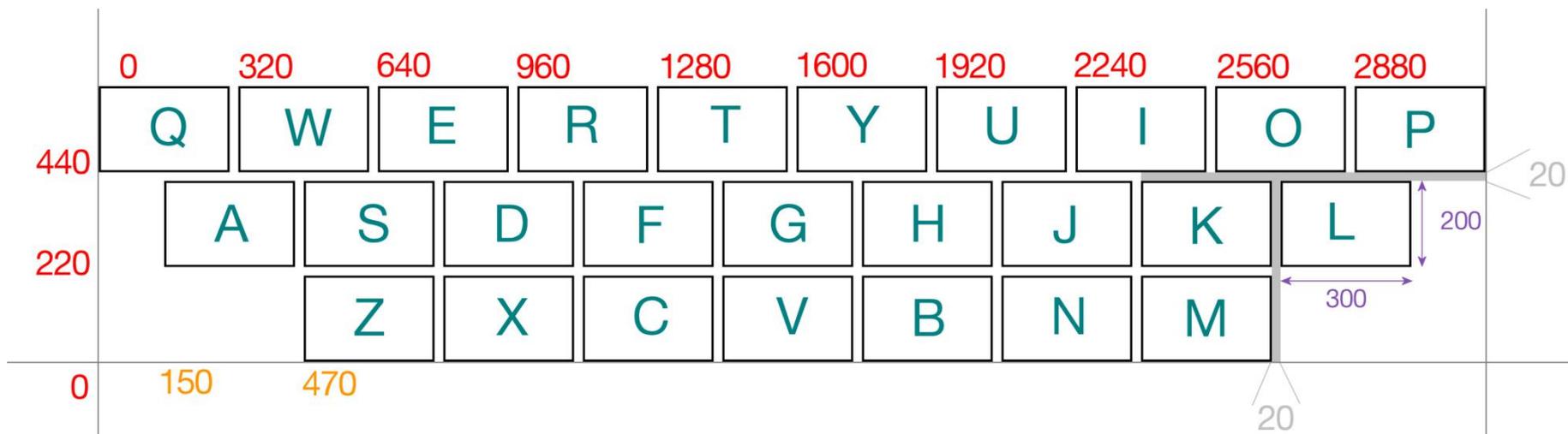
- Svaki pravokutnik je širine 300 te visine 200.
- Između svaka dva pravokutnika u istom 'redu' se nalazi horizontalni razmak od točno 20.
- Između dva susjedna reda je vertikalni razmak od točno 20.
- Donji lijevi kut pravokutnika koji sadrži slovo 'Q' je na koordinatama (0, 440).
- Donji lijevi kut pravokutnika koji sadrži slovo 'A' je na koordinatama (150, 220).
- Donji lijevi kut pravokutnika koji sadrži slovo 'Z' je na koordinatama (470, 0).

# Zadatak: Tipke



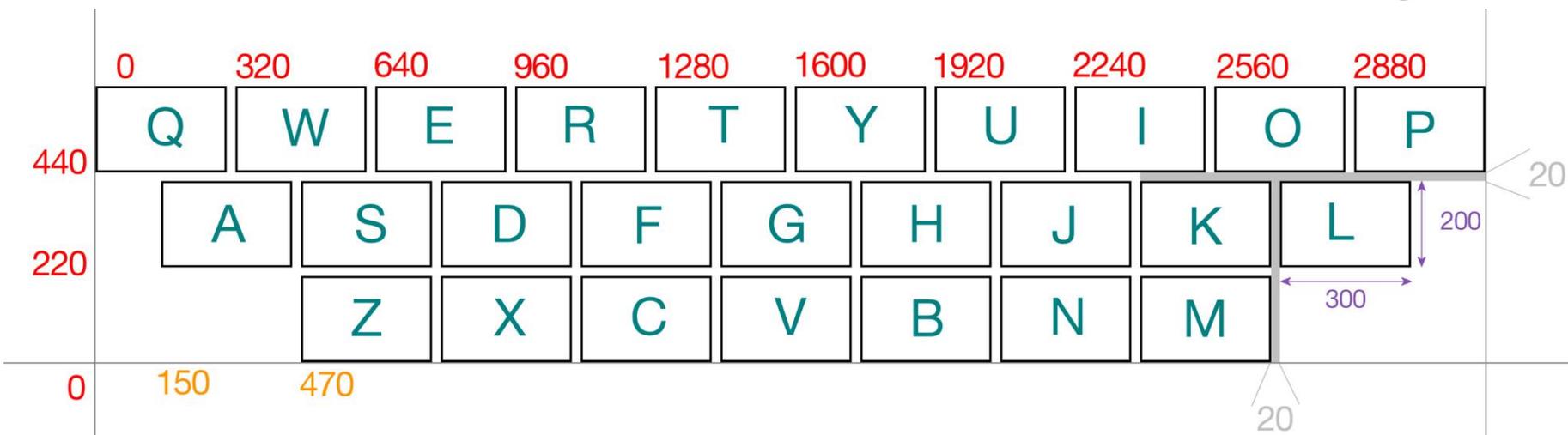
Na temelju ovih pravila i rasporeda tipaka danog u slici gore, mogu se odrediti koordinate pravokutnika za svako slovo. Tako su, na primjer, donji-lijevi i gornji-desni kutovi pravokutnika koji sadrži slovo 'J' jednaki  $(2070, 220)$  i  $(2370, 420)$  redom. *Središte* pravokutnika definiramo kao polovište dužine između dva suprotna kuta pravokutnika. Tako je, na primjer, središte pravokutnika koji sadrži slovo 'J' točka  $(2220, 320)$ . Primijetite da kutovi svih pravokutnika imaju parne  $X$  i  $Y$  koordinate, pa stoga središte svakog pravokutnika ima cjelobrojne koordinate.

# Zadatak: Tipke



Napiši program koji će za N točaka, na temelju njihovih koordinata, odrediti koje je slovo korisnik stisnuo te ispisati ili nađeno slovo ili znak '?' ako točka s koordinatama ne odgovara niti jednom slovu.

# Zadatak: Tipke



ULAZ:

1

1463 378

IZLAZ:

G

ULAZ:

5

2432 173

302 247

982 514

2422 315

2712 624

IZLAZ:

MARKO

ULAZ:

4

2383 240

328 224

1276 513

684 521

IZLAZ:

?A?E

Vrijeme

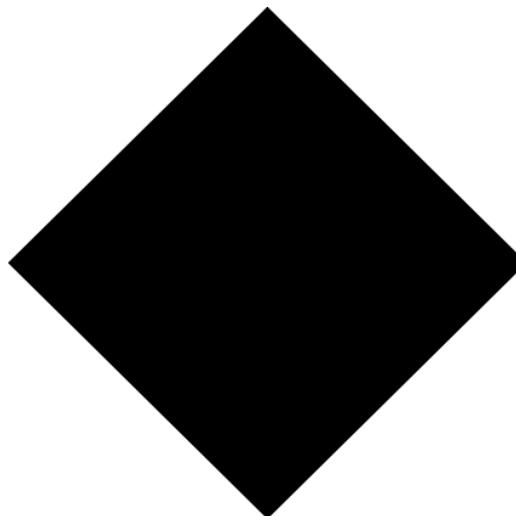


# Zadatak: Kvadrat



Marica voli crtati geometrijske likove na računalu. Međutim, dosadili su joj svi programi za crtanje, pa je odlučila koristiti programe za obradu teksta.

Odlučila je uz pomoć znakova '\*' nacrtati kvadrat, koji je zarotiran za 45 stupnjeva, kao na slici:



# Zadatak: Kvadrat



S obzirom da joj je cijeli svijet programa za obradu teksta nepoznat, potrebna joj je pomoć.

Napišite program koji će Marici 'iscrtati' zadani kvadrat u tekstualnom obliku. Marica će zadati broj N (neparan broj), koji označava duljinu dijagonale kvadrata. Vaš zadatak je iscrtati zadani kvadrat uz pomoć znakova '\*'.

ULAZ:  
3

IZLAZ:  
\*  
\*\*\*  
\*

ULAZ:  
5

IZLAZ:  
\*  
\*\*\*  
\*\*\*\*\*  
\*\*\*  
\*

ULAZ:  
7

IZLAZ:  
\*  
\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*  
\*

Vrijeme

