



**ДРУШТВО
МАТЕМАТИЧАРА СРБИЈЕ**

ПРОГРАМ ЗА ТАКМИЧЕЊЕ ИЗ ПРОГРАМИРАЊА ЗА УЧЕНИКЕ ОСНОВНИХ ШКОЛА

Циљ овог програма је да се:

- 1) одреди оквир у коме ће се састављати и бирати такмичарски задаци;
- 2) такмичарима и онима који планирају и реализују припреме за такмичење предочи у ком оквиру могу да се очекују задаци на такмичењима.

Циљ овог програма није да сам по себи представља програм припрема за такмичење, већ само да одреди елементе које би свакако требало обухватити у припреми за такмичење. Познавање наведених тема може да помогне у фокусању припрема на одређена подручја и знања.

На такмичењу, учесници се срећу са задацима који траже промишљање о датом проблему. Такође такмичари се срећу и са начинима како осмишљено решење записати у одабраном програмском језику.

Сви задаци који се задају на такмичењима су направљени да буду алгоритамски по природи и да се решавају у форми конзолних апликација. Формат улазних и излазних података су прецизно задати. Улазни подаци се учитавају са стандардног улаза, и они ће у тест примерима за евалуацију бити у описаном облику. Коректност улазних података није потребно проверавати у програму. Такмичарев програм мора исписивати излазне податке искључиво у траженом формату на стандардни излаз.

По питању елемената програмског језика овде описана очекивана знања и вештине су довољни за решавање свих задатака, што не искључује да одређена додатна знања и вештине у вези елемената програмског језика могу бити корисна за такмичара.

По питању техника програмирања и алгоритама се оставља могућност да тежи задаци буду тешки већини такмичара, да би се постигло адекватно рангирање између најбољих такмичара, а то се посебно односи на више нивое такмичења.

1.1 ОЧЕКИВАНО ЗНАЊЕ ПРЕМА НИВОИМА ТАКМИЧЕЊА

У табели 1 је приказан очекивани ниво знања према нивоима такмичења.

Табела 1. Очекивано знање према нивоима такмичења

Ниво такмичења

● – може се очекивати

◐ – може се очекивати у најтежем задатку

	I катег.- општин.	I катег.- окружно	I катег.- државно	II катег.- општин.	II катег.- окружно	II катег.- државно	СИО
Петље	◐	●	●	●	●	●	●
Низови		◐	●	●	●	●	●
Бројеви у покретном зарезу				●	●	●	●
Матрице			◐	◐	●	●	●
Напредне технике програмирања I			◐		◐	●	●
Напредне технике програмирања II						◐	●

1.2 ПРОМЕНЉИВЕ И ИЗРАЗИ ЗА ОСНОВНЕ ТИПОВЕ ПОДАТАКА

Такмичар зна да користи променљиве, доделу вредности и изразе у којима се појављују целобројне вредности, вредности у покретном зарезу, карактери, стрингови и истинитосне вредности.

Такмичар зна да користи целобројне типове чији опсег одговара 32-битном означеном целом броју и разуме разлику и рад са ширим и ужим регистром.

Такмичар зна да користи барем један тип за бројеве у покретном зарезу који може бити једноструке или двоструке прецизности.

Такмичар зна како се у програмском језику изводе следеће операције:

- сабирање, множење и одузимање целих бројева и бројева у покретном зарезу;
- унарни минус, апсолутна вредност и знак целог броја и броја у покретном зарезу;
- целобројно дељење и остатак при дељењу целих бројева;
- дељење бројева у покретном зарезу и остатак при дељењу бројева у покретном зарезу;
- квадратни корен броја у покретном зарезу;
- заокруживање броја у покретном зарезу на цео број и то: наниже, навише и на ближи („floor“, „ceiling“ и „round“);
- конверзије између бројевних типова;
- конверзија стринга у број и броја у стринг коришћењем декадног бројног записа;
- дужина стринга;
- надовезивање стрингова и издвајање подстринга;
- конверзија из карактера у његов целобројни код и обрнуто;
- поређење вредности (мање/веће/једнако) за сваки од типова;
- логичке операције дисјункција („или“), конјункција („и“) и негација.

Уколико изабрани програмски језик нема директну подршку за неке од наведених операција, такмичар треба да зна како одговарајућу обраду да изведе користећи расположиве могућности програмског језика.

Такмичар разуме да операције са бројевима у покретном зарезу најчешће дају приближан резултат, а посебно зна како да правилно пореди бројеве у покретном зарезу и како да конвертује број у покретном зарезу у цео број.

1.3 УЛАЗ И ИЗЛАЗ

Такмичар зна да са стандардног улаза учитава целобројне вредности, вредности у покретном зарезу и стринговске вредности када је свака појединачна вредност записана у посебном реду или када су вредности раздвојене сепаратором (белина, зарез, двотачка,...).

Такмичар зна да на стандардни излаз исписује бројеве, стрингове и карактере и да их при томе распоређује у редове.

Приликом исписивања вредности у покретном зарезу такмичар зна да постави формат исписивања тако да се користи одређен број цифара иза децималне тачке.

1.4 ГРАНАЊА

Такмичар зна да користи контролну структуру гранања „if“ са и без „else“. Такмичар зна да користи гранање у оквиру израза и вишеструко гранање уколико изабрани програмски језик то подржава.

Такмичар зна да имплементира сложене услове који захтевају угњеждена гранања и/или сложене логичке изразе у условима и/или увођење додатних истинитосних променљивих.

1.5 ПЕТЉЕ

Такмичар зна да користи „for“ и „while“ петље.

Такмичар зна да у решењу примени петљу са вишеструким условима за излазак, односно останак у петљи, као и да правилно одреди шта треба да се уради пре уласка у петљу, шта у телу петље, а шта након изласка из петље.

Такмичар зна да у решењу примени итеративне поступке који захтевају вишеструке петље укључујући и комбинацију са вишеструким гранањима.

1.6 ФУНКЦИЈЕ, ПРОЦЕДУРЕ, МЕТОДЕ И ПОТПРОГРАМИ

Такмичар зна како да део кода издвоји у функцију, процедуру, метод, односно потпрограм (термини зависе од програмског језика) како би смањио понављање кода и програм учинио прегледнијим.

1.7 НИЗОВИ

Такмичар зна да користи низовске типове података.

Такмичар зна да секвенцу вредности истог типа учита у низ, односно испише на основу вредности елемената низа.

Такмичар зна да секвенцијално обради елементе низа и да при томе према потреби мења вредности елемената тог низа, поставља елементе другог низа, рачуна статистике попут минимума, максимума, збира, производа, просека и слично.

Такмичар да зна да преписује делове низа укључујући промену редоследа елемената.

Такмичар зна да проверава услове који су изражени као међусобне релације елемената низа (на пример, провера да ли је низ уређен, симетричан,...).

1.8 МАТРИЦЕ

Такмичар зна да користи матрице (дводимензионе низове) било као низове низова или као посебан тип уколико програмски језик то потржава.

Такмичар зна да секвенцу вредности истог типа учита у елементе матрице по врстама или по колонама, односно да испише на основу вредности елемената матрице.

Такмичар зна да секвенцијално обрађује елементе матрице (по врстама, по колонама и сл.) и да при томе према потреби мења вредности елемената матрице, поставља елементе друге матрице, рачуна статистике (попут минимума, максимума, збира, производа, просека и слично) по врстама или по колонама и резултате смешта у низ или их непосредно даље обрађује.

Такмичар зна да преписује правоугаоне делове матрице укључујући промену редоследа елемената.

Такмичар зна да проверава услове који су изражени као међусобне релације елемената матрице (на пример да ли је матрица симетрична).

1.9 ОСНОВНЕ ТЕХНИКЕ ПРОГРАМИРАЊА

Такмичар зна да решава задатке који се заснивају на опису реалног или замишљеног догађаја, разумевању описаних односа и примени елементарних физичких и математичких законитости.

Такмичар зна да решава задатке у којима је потребно осмишљавање алгорита примерене сложености.

Такмичар влада техником коришћења целобројног дељења, односно дељења са заокруживањем и остатка при дељењу у решавању проблема који су засновани на неопходности заокруживања бројева.

Такмичар зна да у решењу реализује релације и операције из теорије бројева као што су одређивање простих бројева, растављање на просте бројеве, највећи заједнички делилац и слично.

Такмичар влада техником растављања броја на цифре декадног записа и израчунавања броја када се могу одредити цифре декадног записа.

Такмичар влада техником формирања римског записа броја и израчунавања броја када се могу одредити цифре римског записа.

Такмичар разуме шта значи када су одређени објекти распоређени у врсте и колоне или када је правоугаоник подељен на једнаке квадрате (поља) попут шаховске табле и у тако формулисаним задацима:

- зна да користи редне бројеве врста и колона у одређивању релација између положаја објеката укључујући аналогију кретања шаховских фигура;
- разуме дефинисање правоугаоног подручја (скупа поља, односно објеката) и подручја другог карактеристичног облика и зна да решава проблеме засноване на својствима и

односима таквих подручја укључујући и одређивање броја поља у троугаоним формама (где се користи рачунање збира узастопних бројева);

- зна да у решењу реализује симулацију кретања на основу секвенце вредности које описују потезе према задатим правилима.

Такмичар зна да у решењу реализује сортирање низа алгоритмом који има квадратну сложеност или мању, као и да користи сортирање у решавању проблема.

Такмичар зна да у решењу реализује линеарно претраживање низа или матрице.

Такмичар зна да у решењу реализује рачунање са временима и датумима.

Такмичар зна да решава задатке у којима се користе правила првенста пролаза у раскрсници, уколико су та правила објашњена у задатку.

Такмичар зна да решава задатке у којима је потребно одредити односе тачака, дужи и правоугаоника коме су ивице паралелне осама координатног система у равни.

1.10 НАПРЕДНЕ ТЕХНИКЕ ПРОГРАМИРАЊА I

Такмичар зна да оцени сложеност алгоритма који примењује, како по питању времена извршавања, тако и по питању заузимања меморије.

Такмичар детаљно познаје опсег вредности, прецизност, заузеће меморије и перформансе основних операција за основне типове података које нуди програмски језик и на основу тога уме да изабере одговарајући тип.

Такмичар зна да у решењу реализује ефикасне алгоритме сортирања и претраге.

Такмичар влада техником рекурзивног свођења на потпроблеме.

Такмичар влада техником рекурзивне претраге укључујући претрагу по дубини и по ширини, као и технику претраге са враћањем (backtracking).

Такмичар зна да у решењу реализује алгоритме из области комбинаторике.

Такмичар зна да решава проблеме у којима је потребно одредити односе тачака, дужи, троуглова и правоугаоника у равни.

Такмичар зна да решава проблеме у којима се појављује структура графа у улазним подацима уколико се алгоритми обраде графова свде на непосредну примену претходно наведених техника и алгоритама.

Такмичар зна да у решењу реализује алгоритам бинарне претраге.

1.11 НАПРЕДНЕ ТЕХНИКЕ ПРОГРАМИРАЊА II

Такмичар влада техником динамичког програмирања и техником мемоизације рекурзивних функција.

Такмичар зна да у решењу реализује специјализоване алгоритме за обраду графова.

Такмичар зна да у решењу реализује геометријске алгоритме који заснивају на односима дужи, тачака, полигона и полиедара.

Такмичар зна да у решењу реализује рачунање са великим бројевима.

Такмичар зна да решава проблеме у којима треба применити идеје из познатих напредних алгоритама сортирања и претраге.