



- Сваки задатак носи максимално 50 бодова.
- На почетку сваког задатка уносе се потребне улазне вредности.
- Сваки задатак има 5 излаза који се рачунају на основу улазних вредности унетих на почетку.
- Сваки излаз у задатку носи 10 бодова ако је тачан, 0 ако је нетачан.
- Такмичари су обавезни да **ЈАСНО** назначе излазе (погледати слику на примерима). Ако излаз није јасно назначен такмичар добија 0 поена.
- Ако задатак престане да ради због грешке у било ком тренутку, такмичару се бодују излази који су до тог тренутка видљиви на екрану.
- Такмичење траје 180 минута.

Задатак 1

Потребно је реализовати 3D матрицу целих бројева на следећи начин. На самом почетку се уноси дужина матрице $1 < N < 20$. Провера уноса обавезна. Вредности у пољима матрице попунити тако да вредност представља суму индекса, ако је елемент на индексима (1,1,1) онда вредност у том пољу треба бити $1+1+1=3$. Програм ради тако што се на почетку једном унесе **N** (број елемената), након тога се могу задавати наредбе, програм се завршава када се унесе наредба **QUIT**. Програм током свог рада мора бити неосетљив на величину слова наредби, то значи да се наредбе могу уносити малим и великим словима. Индексирање креће од 0.

Након тога је потребно реализовати следеће операције:

- 1.1 Приказати вредности које се након иницијализације налазе у матрици.
- 1.2 Потребно је омогућити операцију **UPDATE** над матрицом која би радила тако што би ажурирала вредност на задатим индексима i,j,k и постављала нову вредност. Наредба се уноси из терминала и изгледала би **UPDATE 2,2,2 -1**. Приказати вредности које се након промене налазе у матрици.
- 1.3 Потребно је омогућити операцију **SUM 1,1,1 3,3,3** ова наредба приказује на конзоли суму елемента која се налази на индексима у том опсегу укључујући и тај опсег.
- 1.4 Операција **NEW** формира нову матрицу која је истих димензија као и почетна с тим што су вредности елемената такве да представљају збир суседних елемената из почетне матрице. Суседни елементи су они који се само разликују за по један индекс од тренутне, уколико се ради о елементу који се налази на првом реду онда му је сусед и елемент из последњег реда, важи за све димензије. **(Сматрати да елементи по дијагонали нису суседни)**. Након формирања матрице потребно је новоформирану матрицу исписати.
- 1.5 Додати операцију **MOVE** која се уноси у формату **MOVE 1,1,1 0,1,1**. Ова операција треба да помери (shift) елементе који се налазе у истој x,y,z оси као и елемент чији су индекси задати за одређени број места по x,y,z оси. Померање се врши над иницијалном матрицом. Померање елемената је могуће по свим осама у плус и минус страну (вредности за померање по местима могу бити позитивне и негативне). Након завршеног померања исписати матрицу.



Задатак 2

Унети стринг **s1** величине ($2 < N < 30$) карактера који се састоји од малих слова, великих слова енглеске абецедe и цифара. Претпоставка је да је стринг исправно унет, не треба проверавати да ли је унос исправан. **N** је број који се не уноси.

- 2.1 Потребно је на основу карактера из низа **s1** формирати низ стрингова **brojeviNiz** који представљају ASCII вредност карактера тако да они буду представљени као **троцифрени** ('a'='097', 'z'='122'). Низу **brojeviNiz** изменити елементе тако да се на елементима који имају парни индекс између сваке две цифре уметне број који представља аритметичку средину цифара заокружену на већи број (средина је: $(0+9+7)/3=5.3 \sim 6$, тако да уместо '097' треба да пише '06967'). Приказати само елементе који се налазе на парним индексима у новоформираном низу.
- 2.2 Одредити да ли је стринг **s1** правилан или није, сматра се да је стринг исправан ако се сви знакови појаве исти број пута. Такође се сматра да је исправан ако се може уклонити само један карактер на некој позицији, а преостали се појаве исти број пута. Исписати да ли је стринг **s1** правилан или није.
- 2.3 Потребно је у нови стринг **s2** из стринга **s1** пребацити само слова **A, C, T и G**. Приликом уписа у нови стринг се преписују ови карактери заменарујући величину слова док се у новом стрингу **s2** записују као велика слова. Приказати **s2**.
- 2.4 Тако направљен стринг се сматра **геном**. Ген се сматра стабилним ако је дужине дељиве са **4** и ако је учесталост појављивања сваког фактора **n/4** (ако је ген дужине 8 карактера, **сваки** фактор (**A, C, T и G**) се мора појавити два пута). Исписати да ли је **s2** стабилан ген. Исписати учесталост појављивања сваког слова односно колико пута се које слово појављује.
- 2.5 Потребно је наћи дужину најмањег подниза који се може који додати на крај низа како би се ген направио стабилним. Приказати најкраћи подниз за додавање, сортиран по абецеди растуће.

Задатак 3

Направљен је нови тржни центар који се много свидео једном лопову. Његов циљ је да за једну ноћ у што краћем року опљачка све радње у том тржном центру. Оно што лопов не зна је да сваку ноћ долази полицајац који се шета кроз спратове тржног центра.

Лопов је направио мапу тржног центра јер тржни центар има 10 спратова и на сваком спрату има 18 радњи, а на ивицама спратова су степенице. Ради лакшег рачунања, лопов је одлучио да координате мапе почињу од 0 и да се поклапају са редовима и колонама матрице (координате 0,0 би биле први елемент матрице у горњем левом углу).



	y-osa																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
x-osa																				
0																				
1																				
2																				
3																				
4																				
5																				
6																				
7																				
8																				
9																				
10																				
11																				
12																				
13																				
14																				
15																				
16																				
17																				
18																				
19																				

Одједном се чује аларм! Полицајац се налази на Хр, Ур позицији и жури ка најближим степеницама, у журби прелази по два поља док не дође до степеница. Кад дође до степеница креће се горе-доле не би ли видео лопова на том спрату. Ако може, прво се пење на спрат изнад. **У једном потезу иде се степеницама по један спрат.**

Лопов је на позицији Хl, Уl и креће редом да пљачка радње једну по једну (прво креће на страну где има више радњи на спрату). Кад дође до степеница, ако може спушта се на спрат испод и пљачка радње на том спрату. Прескаче спрат на коме је опљачкао све радње.

Кад полицајац види лопова на истом спрату, креће се по два поља ка њему у једном потезу да би га ухватио, док лопов бежи ка другим степеницама по једно поље.

Два излаза из тржног центра се налазу на подножју степеница. Степенице су обележене са **S**, а степенице за излазом са **E**.

Ако лопов опљачка све радње излази из тржног центра на најближем излазу.

Улазни параметри су:

- Хр, Ур – Почетне координате локације полицајца унете као индекси матрице
- Хl, Уl – Почетне координате локације лопова унете као индекси матрице

И уносе се у формату

Хр,Ур

Хl,Уl

Урадити следеће задатке:

- 3.1 Приказати матрицу тржног центра као на тест примеру. У сваком пољу треба да буде карактер „R“ што представља радњу, „S“ што представља степенице и „E“ што представља излаз.
- 3.2 Приказати матрицу тржног центра са почетном позицијом полицајца и лопова. Символ за полицајца је „P“ а симбол за лопова је „L“.
- 3.3 У матрици приказати поља кроз која је лопов прошао са симболом „-“ кад се игра заврши.
- 3.4 Ако полицајац ухвати лопова, то место означити са „X“. Исписати координате радње у којој га је полицајац ухватио. Ако лопов побегне, исписати поруку да су све радње опљачкане.
- 3.5 Приказати успешност пљачке до хватања лопова или до бега. Успешност представља проценат опљачканих радњи не узимајући у обзир радњу у којој је лопов ухваћен, али



узимајући у обзир рању из које је кренуо. Приказати успешност као децимални број заокружен на 2 децимале.