



## Olimpiada Internazionale ne Informatike 2012

23-30 Shtator 2012

Sirmione - Montichiari, Italy

Detyrat e Konkursit, dita 2: Shkenca dhe arti i Leonardos

---

## Darka e fundit

---

Leonardo ishte shume aktiv ndersa punonte me Darken e Fundit, piktura murale e tij me e njohur: nje nga detyrat e tij ditore ishte te vendoste se cilat nga ngjyrat e posterit te perdorte ate dite. Ai kishte nevojte per shume ngjyra por mund te mbante nje numer te limituar ne tribunen e tij. Ndhimesi i tij kishte si detyre ndermjet gjerave te tjera, te hipte ne tribunen e tij per ti dhene ngjyrat dhe pastaj te zbriste poshte dhe ti vendoste ne nje raft te pershtashem poshte.

Ne kete detyre, duhet te shkruani dy programe te ndara per te ndihmuar ndihmesin. Programi i pare do te marre instruksionet e Leonardos (nje sekuence me ngjyrat per te cilat Leonardo do te kete nevojte gjate dites ) dhe te krijojte nje string te shkurter me bite, te quajtur keshille. Gjate dites, ndihmesi nuk do te kete akses ne kerkesat e Leonardos, vetem ne keshillat e krijuara nga programi juaj i pare. Programi i dyte do te marre keshillen dhe do te procesoje kerkesat e Leonardos ne menyren online (p.sh nje ne nje kohe). Ky program do te jete i afte te kuptojte cfare domethenieje ka keshilla, ta perdore ate per te bere zgjedhjen optimale. Cdo gje eshte shpjeguar me me shume detaje.

### Levizja e ngjyrave ndermjet raftit dhe tribunes

Do te konsiderojme nje skenar te thjeshtezuar. Supozoni se ka  $N$  ngjyra te numeruara nga  $0$  ne  $N - 1$  dhe cdo dite Leonardo pyet ndihmesin per nje ngjyre te re ekzaktesisht  $N$  here. Le te jete  $C$  sekuenca e kerkeses se  $N$  ngjyrave te bere nga Leonardo. Mund ta mendojme  $C$  si sekuencen e  $N$  numrave, secili prej tyre eshte ne intervalin  $0$  dhe  $N - 1$  ( $N-1$  perfshihet). Vini re se disa ngjyra mund te mos jene ne  $C$ , dhe te tjerat te shfaqen shume here.

Tribuna eshte perhere plot dhe permбан  $K$  nga  $N$  ngjyrat, ku  $K < N$ . Fillimisht, tribuna permбан ngjyrat nga  $0$  ne  $K - 1$ , perfshire.

Ndhimesi proceson kerkesat e Leonardos nje nga nje. Sa here qe ngjyra e kerkuar ndodhet ne tribune, ndihmesi mund te pushoje. Perndryshe, ai duhet ta marre ngjyren e kerkuar nga rafti dhe ta levize drejt tribunes. Sigurisht qe ne tribune nuk ka vend per ngjyren e re, prandaj ndihmesi duhet te zgjedhe nje ngjyre nga tribuna, ta marre dhe ta rivendose ne raft.

### Strategjia optimale e Leonardos

Ndhimesi don te pushoje sa me shume here qe te jete e mundshme. Numri i kerkesave per te cilat mund te pushoje varet nga zgjedhjet e tij te bera gjate procesit. Saktesisht, sa here qe ndihmesit i duhet te heqe nje ngjyre nga tribuna, zgjedhje te ndryshme mund te shkaktojne rezultate te ndryshme ne te ardhmen. Leonardo i shpjegon atij se si te arrije qellimin e tij duke njohur  $C$ . Zgjedhja me e mire per ngjyren qe do te hiqet nga tribuna merret nga ekzaminimi i ngjyrave qe ndodhen tashme ne tribune dhe kerkesave te mbetura per ngjyra ne  $C$ . Nje ngjyre duhet te zgjidhet ndermjet atyre qe ndodhen ne tribune sipas rregullave te meposhtme:

- Nese nje ngjyre qe ndodhet ne tribune nuk do te nevojitet asnjehere ne te ardhmen, ndihmesi duhet ta heqe ate ngjyre nga tribuna.
- Perndryshe, ngjyra e hequr nga tribuna duhet te jete ajo qe do te nevojitet me vone se te tjerat ne te ardhmen. (Pra, per cdo ngjyre qe ndodhet ne tribune do te gjehet shfaqja e pare ne te ardhmen. Ngjyra qe do te levizet per ne raft do te jete ajo qe do te nevojitet me vone.)

Mund te provohet qe duke perdorur strategjine e Leonardos, ndihmesi do te pushoje sa me shume qe te jete e mundur.

### Shembulli 1

Le te jete  $N = 4$ , pra kemi 4 ngjyra ( te numeruara nga  $0$  ne  $3$ ) dhe 4 kerkesa. Le te jete sekuenca e kerkesave  $C = (2, 0, 3, 0)$ . Gjithashtu supozoni se  $K = 2$ . Pra, Leonardo ka nje tribune te afte te mbaje 2 ngjyra ne te njejten kohe. Sic eshte deklaruar me lart, tribuna permбан numrat  $0$  dhe  $1$ . Permbajtja e tribunes do te shkruhet:  $[0, 1]$ . Nje menyre e mundshme qe ndihmesi mund ta trajtojte kerkesen eshte si me poshte.

- Ngjyra e pare e kerkuar (numri 2) nuk ndodhet ne tribune. Ndhimesi e vendos atje dhe vendos te levize ngjyren 1. Tribuna korrente eshte [0, 2].
- Ngjyra tjetere e kerkuar (numri 0) ndodhet tashme ne tribune, prandaj ndihmesi mund te pushoje.
- Per kerkesen e trete (numri 3), ndihmesi leviz ngjyren 0, duke ndryshuar tribunen ne [3, 2].
- Me ne fund, ngjyra e fundit e kerkuar (numri 0) duhet te merret nga rafti dhe te vendoset ne tribune. Ndhimesi vendos te heqe ngjyren 2 dhe tribuna behet tani [3, 0].

Vini re ne shembullin e mesiperme qe ndihmesi nuk ndoqi strategjine optimale te Leonardos. Strategjia optimale do te hiqte ngjyren 2 ne hapin e trete, keshtu ndihmesi do te pushonte ne hapin e fundit.

### Strategjia e ndihmesit kur memorja e tij eshte e limituar

Ne mengjes ndihmesi pyet Leonardon te shkruaje C ne nje cope letreje, ne menyre qe ai te gjeje dhe te ndjeke strategjine optimale. Megjithate, Leonardo deshiron qe ta mbaje tekniken e punes se tij sekrete, prandaj refuzon qe ndihmesi te kete letren. Ai i lejon ndihmesit qe ta lexoje C dhe te perpiqet qe ta mbaje mend ate.

Fatkeqesisht, memorja e ndihmesit eshte shume e keqe. Ai mund te mbaje mend deri ne M bite. Ne pergjithesi, kjo gje do ta pengonte ate qe te ndertonte te tere sekuencen e C. Prandaj, ndihmesi llogarit sekuencen e biteve qe mund te maje mend. Do ta quajme kete sekuence, sekuenca e keshilles dhe do ta shenojme me A.

### Shembull 2

Ne mengjes, ndihmesi mund te marre letren e Leonardos me sekuencen C, ta lexoje sekuencen dhe te beje te gjitha zgjedhjet e nevojshme. Nje gje qe mund te zgjedhe te beje eshte te ekzaminoje gjendjen e tribunes pas cdo kerrese. Per shembull, kur perdor strategjine(sub-optimale) te dhene ne shembullin 1, sekuenca e gjendjeve te tribunes do te jete [0, 2], [0, 2], [3, 2], [3, 0]. ( Kujtoni se gjendja fillestare e tribunes eshte [0, 1].)

Tani supozoni se kemi  $M = 16$ , keshtu ndihmesi eshte i afte te mbaje mend deri ne 16 bit me informacion. Nese  $N = 4$ , mund te ruajme cdo ngjyre duke perdorur 2 bite. Prandaj 16 bite jane te mjaftueshme per te ruajtur sekuencen e mesiperme te gjendjeve te tribunes. Prandaj ndihmesi llogarit sekuencen e meposhtme te keshilles:  $A = (0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0)$ .

Me vone gjate dites, ndihmesi mund te dekodoje kete sekuence keshille dhe ta perdore ate per te bere zgjedhjet e tij.

(Sigurisht, me  $M = 16$  ndihmesi mundet qe gjithashtu te zgjedhe te kujtoje te tere sekuencen e C, duke perdorur vetem 8 nga 16 bitet ne dispozicion. Ne kete shembull donim te ilustronim qe ai mund te kete opsione te tjera, pa dhene ndonje zgjidhje te mire.)

## Detyra

Duhet te shkruani dy programe te vecante ne te njejten gjuhe programimi. Keto dy programe do te ekzekutohen ne menyre sekuenciale, pa qene e mundur qe te komunikojne me njeri-tjerin gjate kohes se ekzekutimit te tyre.

Programi i pare do te jete ai qe perdoret nga ndihmesi ne mengjes. Programit do ti jepet nje sekuence C, dhe ai duhet te llogarise nje sekuence A me keshilla.

Programi i dyte do te jete ai qe do te perdoret nga ndihmesi gjate dites. Ky program do te marre ne hyrje nje sekuence A me keshilla, dhe me pas ka per detyre te procesoje sekuencen C me kerkesat e Leonardos. Kerkesat e sekuences C do ti jepen ketij programi nje nga nje. Cdo kerrese duhet te procesohet nga programi perpara se te pranohet kerkesa tjetere.

Me sakte, ne programin e pare duhet te implementoni nje rutine `ComputeAdvice(C, N, K, M)` e cila merr ne input

- nje array C me N numra te plote (ne intervalin 0, ...,  $N - 1$ ),
- Numrin K te ngjyrave ne tribune, dhe
- Numrin e biteve ne dispozicion per keshillimin, M

Ky program duhet te llogarise nje sekuence me keshilla A qe ka jo me shume se M bite. Programi me pas duhet tia komunikojte kete sekuence A sistemit dike therriur sipas rashes per cdo bit te A rutinen e meposhtme:

- `WriteAdvice(B)` — e shton bit B ne fund te sekuences korrente A. (Kjo rutine mund te therrihet jo me shume se M here)

Ne programin e dyte duhet te implementoni nje rutine te vetme `Assist(A, N, K, R)`. Input i kesaj rutine eshte sekuenca e keshillave `A`, numrat e plote `N` dhe `K` sic i perkufizuam me lart, dhe gjatesia aktuale `R` e sekuences `A` ne bite ( $R \leq M$ ). Kjo rutine duhet te ekzekutoje strategjine tuaj te propozuar per ndihmesin, duke perdorur rutinat e meposhtme, te dhena.

- `GetRequest()` — kthen ngjyren e rradhes te kerkuar nga Leonardo. (Nuk zbulohet asnje informacion mbi kerkesat e ardhshme.)
- `PutBack(T)` — kthen ngjyren `T` nga tribuna ne raft. Kete rutine mund ta therrisni vetem me nje `T` qe eshte nje nga ngjyrat qe ndodhen ne kete moment ne tribune.

Kur ekzekutohet rutina `JuajAssist` ajo duhet te therrase `GetRequest` sakesisht `N` here, nje per cd okerkese te Leonardos. Pas cdo thirrje te `GetRequest`, nese ngjyra e kthyer nuk ndodhet ne tribune duhet te therrisni gjithashtu `PutBack(T)` me `T` e zgjedhur nga `Ju`. Ne te kundert, nuk duhet te therritni `PutBack`. Nese e therrisni gabimisht rutinen do te konsiderohet si gabim dhe do te shkaktoje perfundimin e programit `Tuaj`. Fillimisht ne tribune ndodhen ngjyrat nga `0` ne `K - 1`, perfshire.

Nje test do te konsiderohet i zgjidhur nese rutinat `Tuaja` zbatojne te gjitha kufizimet e vendosuar me lart dhe numri total i therritjeve te eshte sakesisht i barabarte me strategjine optimal te Leonardos. Nese ka disa strategji qe kapin te njejtin numer thirrjesh te `PutBack`, program mund te perdore njerin prej tyre

### Shembull 3

Ne vazhdimesi te shembullit 2, le te supozojme se ne `ComputeAdvice` llogaritem `A = (0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0)`. Ne menyre qe tua komunikojme kete sistemit duhet te beni therritjet e meposhtme: `WriteAdvice(0), WriteAdvice(0), WriteAdvice(1), WriteAdvice(0), WriteAdvice(0), WriteAdvice(0), WriteAdvice(1), WriteAdvice(0), WriteAdvice(1), WriteAdvice(1), WriteAdvice(1), WriteAdvice(1), WriteAdvice(0), WriteAdvice(1), WriteAdvice(1), WriteAdvice(0), WriteAdvice(0)`.

Rutina `Juaj` e dyte `Assist` duhet te ekzekutohet duke marre sekuencen e mesiperme `A` dhe vlerat `N = 4`, `K = 2`, dhe `R = 16`. Runtina `Assist` me pas, duhet te beje sakesisht `N = 4` thirrje tek `GetRequest`. Gjithashtu pas disa kerkesave, `Assist` do te duhet te therrase `PutBack(T)` me nje zgjedhje te pershtatshme te `T`.

Tabela me poshte tregon nje eskeuence me thirrje qe i korrespondon zgjedhjeve te bera ne shembullin 1. Vija tregon qe nuk therritet `PutBack`.

<code>GetRequest()</code>	Action
2	<code>PutBack(1)</code>
0	-
3	<code>PutBack(0)</code>
0	<code>PutBack(2)</code>

### Nendetyra 1 [8 pike]

---

- $N \leq 5\,000$ .
- Maksimalisht M mund te jete  $M = 65\,000$  bits.

### Nendetyra 2 [9 pike]

---

- $N \leq 100\,000$ .
- Maksimalisht M mund te jete  $M = 2\,000\,000$  bits.

### Nendetyra 3 [9 pike]

---

- $N \leq 100\,000$ .
- $K \leq 25\,000$ .
- Maksimalisht M mund te jete  $M = 1\,500\,000$  bits.

### Nendetyra 4 [35 pike]

---

- $N \leq 5\,000$ .
- Maksimalisht M mund te jete  $M = 10\,000$  bits.

### Nendetyra 5 [deri ne 39 pike]

---

- $N \leq 100\,000$ .
- $K \leq 25\,000$ .
- Maksimalisht M mund te jete  $M = 1\,800\,000$  bits.

Piket qe do te merrni per kete nendetyre varen nga gjatesia R e keshillave qe komunikon program JuajThe. Me konkretisht, nese  $R_{\max}$  eshte maksimumi (ne te gjithë rastet test) I gjatesise se sekuences se keshillave te prodhuar ngfa rutina Juaj ComputeAdvice, piket Tuaja do te jene:

- 39 pike nese  $R_{\max} \leq 200\,000$ ;
- $39 \cdot (1\,800\,000 - R_{\max}) / 1\,600\,000$  pikr nese  $200\,000 < R_{\max} < 1\,800\,000$ ;
- 0 points if  $R_{\max} \geq 1\,800\,000$ .

### Detaje te implementimit

---

Duhet te ngarkoni dy skedare ne te njeften gjihe programimi.

Skedari i pare quhet `advisor.c`, `advisor.cpp` ose `advisor.pas`. Ky skedar duhet te permbaje rutinen `ComputeAdvice` dhe te mund te therrase rutinen `WriteAdvice`. Skedari i dyte quhet `assistant.c`, `assistant.cpp` ose `assistant.pas`. Ky skedar duhet te permbaje rutinen `Assist` dhe tem und te therrase rutinat `GetRequest` dhe `PutBack`.

Prototipet e rutinave jane me poshte: The signatures for all the routines follow.

#### Programme ne C/C++

```
void ComputeAdvice(int *C, int N, int K, int M);
void WriteAdvice(unsigned char a);
void Assist(unsigned char *A, int N, int K, int R);
void PutBack(int T);
int GetRequest();
```

#### Programme ne Pascal

```
procedure ComputeAdvice(var C : array of LongInt; N, K, M : LongInt);
procedure WriteAdvice(a : Boolean);
procedure Assist(var A : array of Byte; N, K, R : LongInt);
procedure PutBack(T : LongInt);
function GetRequest : LongInt;
```

Rutinat duhet te funksionojne sic u pershkrua me lart. Sigurisht qe jeni te lire te shkruani rutina te tjera pe perdorim te brendshem. Per programe ne C/C++, rutinat tuaja duhet te deklarohen si statike, pasi sample grader-i do ti lidhe ato bashke. Nuk duhet te nderveproni me standard input/output, apo me ndonje skedar tjeter.

#### Programme ne C/C++

Ne fillim te zgjidhjes Tuaj, duhet te perfshinin skedarin `advisor.h` dhe `assistant.h`, perkatesisht, tek `advisor` dhe tek `ndhimesi`. Kjo behet duke perfhsire nje rresht kodi ne fillim te programit Tuaj:

```
#include "advisor.h"
```

ose

```
#include "assistant.h"
```

Dy skedaret `advisor.h` dhe `assistant.h` do d'Ju behen disponibel ne nje direktori Brenda ambientit Tuaj te konkurimit dhe do te ofrohen gjithashtu edhe nga nderfaqja web. Gjithashtu do t'Ju jepet ne te njeitin menyre kod dhe skripte per te kompilur dhe testuar zgjidhjen Tuaj. Gjithashtu, pasi te keni kopjuar zgjidhjen tuaj ne direktorine me keto skripteduhet te ekzekutoni `compile_c.sh` ose `compile_cpp.sh` (ne varesi te gjuhes qe po programoni).

#### Program ne Pascal

Duhet te perdorni unit `advisorlib` dhe `assistantlib`, perkatesisht, tek keshiluesi dhe tek ndihmesi. Kjo behet duke shtuar nje rresht kodi ne fillim te programit:

```
uses advisorlib;
```

ose

```
uses assistantlib;
```

Dy skedarte `advisorlib.pas` dhe `assistantlib.pas` do t'ju behen disponibel ne nje direktori Brenda ambientit te konkurimit si edhe nga nderfaqja web. Gjithashtu do t'Ju behen disponibel ne te njejten menyre kod dhe skripte per te testuar zgjidhjen tuaj. Gjithashtu , pasi te keni kopjuar zgjidhjen ne direktorine ku jane keto skripte do t'ju duhet te therrusni `compile_pas.sh`.

### Sample grader-i

Sample grader-i do te pranoje input te formatuar si me poshte:

- rreshti 1: N, K, M;
- rreshtat 2, ..., N + 1: C[i].

Grader-i do te ekzekutoje ne fillim rutinen `ComputeAdvice`. Kjo do te gjeneroje nje skedat `advice.txt`, qe permban bitet e sekuenes me keshilla, te ndara me hapesire dhe qe terminojne me nje 2.

Me pas do te ekzekutoje rutinen `Tuaj Assist`, dhe te gjenroje output ku sejcili rresht eshte ose ne formen "R [number]", ose ne formen "P [number]". Rreshtat e llojit te pare tregojne thirrje te `GetRequest()` dhe pergjigjet e marra. Rreshtat e llojit te dyte tregojne thirrje te `PutBack()` dhe ngjyrat qe jane zgjidhur per tu rikthyer ne raft nga tribuna. Output perfundon me nje rreshte ne formatin "E".

Graderi zyrtar do te punoje ndryshe. Ne menyre te vecante koha e ekzekutimit mund te jete e ndryshme. Ftoheni te perdorni nderfaqen e testit qe te ekzekutoni kodin tuaj me graderin zyrtar.

## Kufizime te kohes dhe memorjes

---

- Limiti i kohes: 7 seconds.
- Limit i Memorjes: 256 MiB.