O’zbekiston Respublikasi Oliy va O’rta Maxsus Ta’lim Vazirligi
Buxoro Davlat Universiteti

Fizika – matematika fakulteti

“Amaliy matematika va axborot texnologiyalari” kafedrasi

5480100-“Amaliy matematika va informatika” ta’lim yo’nalishi bo’yicha bakalavr darajasini olish uchun

Mirzayeva Gulchehra

Mavzu: Dasturlash bo’yicha olimpiada masalalari va ularni yechishga doir uslubiy ko’rsatmalar. (2-qism)

BITIRUV MALAKAVIY ISHI

Ilmiy rahbar _____ katta o’qituvchi Jalolov O.I

“_____” _____2012-y.

BMI “Amaliy matematika va axborot texnologiyalari” kafedrasining 2012-yil _____№ ___ sonli yig’ilishida ko’rib chiqildi va himoyaga ruxsat berildi.

Kafedra mudiri _____dots.I.G.Rasulov Taqrizchi _____

“_____” _____ 2012-y. “_____” _____2012-y.

“Himoyaga ruxsat berildi”
Fakultet dekani _____prof.Sh.M.Mirzayev
“_____” _____ 2012-y.

Buxoro-2012

Mundarija

Kirish 3
I BOB. Internet tizimida olimpiada masalalari va ularni yechish 6
1.1.Dasturlash bo`yicha olimpiadaga oid saytlar 6
1.2.Java dasturlash tilida olimpiada masalalarini yechish va tekshirish 9
Birinchi bob bo`yicha xulosa 26
II BOB. Mavzulashtirilgan olimpiada masalalari 27
2.1.Oddiy matematika(Простая математика) 27
2.2.Uzun sonlar arifmetikasi(Длинная арифметика) 42
2.3.Ikki o`lchovli massivlar(Двумерные массивы) 52
2.4.Satrni tahrirlash(Разбор строк) 69
2.5.Ketma-ketrliklar va ularni saralash(Сортировка и последовательности)82
Ikkinchi bob bo`yicha xulosa 98
Xotima 99
Foydalanilgan adabiyotlar 101

Kirish

Respulikamiz ta’lim tizimidagi asosiy vazifa jahon talablariga mos keluvchi axborot texnologilarini o’qitish jarayoniga qo’llashdan iborat.

O’zbekistonda ta’lim tizimining axborotlashtirilishi xalqaro hamjamiyatda ham tan olindi. Masofaviy ta’limni rivojlantirish bo’yicha bir qator dasturlar ishlab chiqilmoqda. Iqtisodiyot va jamiyatda islohotlarning o’tkazilishi natijasida o’quv jarayonining zahira hajmini keskin oshirish bo’yicha yangi talablar qo’yildi. Hozirgi kunda axborot eng asosiy ishlab chiqaruvchi resurslardan biriga, iqtisodiyot va umuman jamiyatning rivojlanish poydevoriga aylanmoqda.

Yangi axborot – kommunikatsion texnologiyalar hozirgi kunda eng dolzarb mavzulardan biri bo’lib kelmoqda, sababi har bir sohani o’rganish, izlanish va tajriba ortirish uchun turli usullardan foydalanish kerak bo’ladi. Shuning uchun yangi axborot kommunikatsion texnologiyalardan foydalanish maqsadga muvofiqdir.

Respublikamizda olib borilayotgan islohotlarning tarkibida yuqori malakali mutaxasislarning roli benihoyat kattadir. Prezidentimiz ta’kidlaganidek, “Ertangi kun yangicha fikrlay oladigan zamonaviy bilimga ega bo’lgan yuksak malakali mutaxasislarni talab etadi”. Shu sababli xalqimizning boy intelektual merosi va umumbashariy qadriyatlari, zamonaviy madaniyat, iqtisodiyot, fan, texnika va texnologiyalar asosida yuksak mutaxasislar tayyorlash tizimi ishlab chiqildi va jadal sur’atlar bilan hayotga tadbiq etilmoqda.

O’zbekistonda o’qitish texnologiyalarini zamonaviylashtirish, jadallashtirish rivojlangan iqtisodiyotli mamlakatlarga qaraganda yanada dolzarb ahamiyatga ega. Chunki hozirgi kunda milliy ta’lim tizimining salohiyatli tizimli rivojlanishi yanada yuqori pog’onaga ko’tarildi.

Prezidentimiz I.A.Karimov 2001-yil Oliy Majlisning 5-sessiyasida so’zlagan nutqida axborot texnologiyalarni va kompyuterlarni jamiyat hayotiga kishilarning turmush tarziga, o’rta maktab, o’rta maxsus va oliy ta’lim muassasalariga jadallik bilan kirish g’oyasi ilgari surildi. Mamlakatimiz rivojlangan davlatlar qatoridan

mustahkam o’rin egallashi uchun esa zamonaviy kompyuter texnologiyalarini hayotimizning barcha jabhalariga keng joriy etish zarur. Yangicha fikrlaydigan, bozor sharoitlarida muvaffaqiyatli xo’jalik yuritadigan yuksak malakali,chuqur bilimli mutaxassislardan zamonaviy kompyuter texnologiyalarining keng imkoniyatlarini o’zlashtirgan holda ulardan o’z faoliyatlarida samarali foydalana olish talab etiladi.

Hozirgi zamon jamiyati rivojlanish ko’rsatkichlarini aniqlovchi me`yorlardan biri, uning XX asr vositalari bilan ta’minlanish va ularni qo’llash darajasidir. EHM ni samarali qo’llashni amalga oshirish yo’llaridan biri yuqori darajadagi dasturlash tillarini chuqur o’zlashtirish, EHM tizimlari, dasturlash texnologiyalari vositalarini amaliyotga keng qo’llashdan iborat. Shu maqsadda bugungi kunda raqobatbardosh yuqori malakali kadrlarni tayyorlash vazifasi o’rtaga tashlangan.

Bitiruv malakaviy ishi mavzusining dolzarbligi. Talabalar uchun dasturlashga oid olimpiada masalalarini yechish usullarini o’rgatish. Dasturlash bo’yicha olimpiada saytlari bilan ishlashni o’rgatish. Kompyuter texnologiyalaridan foydalanishni o’rgatish va shu bilan birga talabalarni bu sohadagi qiziqishlarini oshirish. Yechilgan masalalarni olimpiadaga oid saytlarga yuborib uning to’g’riligini tekshirish va o’z ustlarida mustaqil ishlashni o’rgatish.

Bitiruv malakaviy ishida tadqiqot ob’yekti va predmeti. Dasturlash tillari. Dasturlash bo’yicha olimpiadaga oid saytlar. Dasturiy ta’minotlar bilan jihozlangan kompyuter xonalari. Kompyuterda masala yechish algoritmlari va dasturlari. Masalalarni bajarishda kompyuter texnologiyalari va eng zamonaviy usullardan foydalanish.

Bitiruv malakaviy ishlarining asosiy maqsadi va vasifalari. Dasturlash bo’yicha olimpiada masalalarini yechishda kerakli saytlardan foydalanish jarayonlarini ko’rsatish. Java dasturlash tili yordamida masalalarni yechish. Natijalarini to’g’riligini kompyuterda maxsus saytlar orqali tekshirish. Talabalar mustaqil shug’ullanishlari uchun uslubiy qo’llanma ko’rinishida tayyorlash.

Talabalarga kompyuterda mustaqil masala yechish usullarini ko'rsatish va talabalarni dasturlashga bo'lgan qiziqishlarini oshirish.

Tadqiqot usuli va uslubiyoti. Masalalarni bajarishda Java dasturlash tili imkoniyatlaridan foydalanish usullarini ko'rsatish va ularni qo'llash. Talabalarga kompyuter orqali masalalar bajarish usul va algoritmlarini o'rgatish bilan birga ularga kompyuterda ishlash texnologiyalarini o'rgatish.

Olingan asosiy natijalar: Talabalarga dasturlash bo'yicha olimpiada masalalarini yechish algoritmini o'rgatish, Java dasturlash tilida dastur tuzishni o'rgatish, olimpiadaga oid saytlar bilan ishlashni o'rgatish va o'zbek tilida uslubiy qo'llanma yaratish.

Bitiruv malakaviy ishidan olingan natijalarning **ilmiy *yangiligi va amaliy ahamiyati:*** Masalalarni bajarishda Java dasturlash tili imkoniyatlarini ko'rsatish. Tejamkor algoritm va dastur yaratishni o'rgatish. Talabalar mustaqil masalalarni bajarib uni kompyuter orqali tekshirishi juda katta ahamiyatga ega.

Tadbiq etish darajasi va iqtisodiy samaradorligi. Qo'llash sohasi. Xulosa va takliflar. Talabalar o'z ustlarida mustaqil ishlashlari va bilim saviyalarini oshirish uchun kompyuter texnologiyalaridan foydalanish g'oyasi taklif etilgan.

Bitiruv malakaviy ishining tarkibi va hajmi quyidagicha : Kirish, 2 ta bob, har bir bobning qisqacha xulosasi, adabiyotlar ro'yxati, xotima va ilovalardan iborat bo'lib jami 99 betga bayon qilingan. I bob internet tizimida olimpiada masalalari va ularni yechishga bag'ishlangan, II bobda mavzulashtirilgan olimpiada masalalari yechimlari bilan keltirilgan.

Kirish qismi 3 betdan iborat, tushuntirish qismi 93 betdan iborat. Bitiruv malakaviy ishida 8 ta adabiyotdan foydalanildi.

I- BOB. Internet tizimida olimpiada masalalari va ularni yechish.

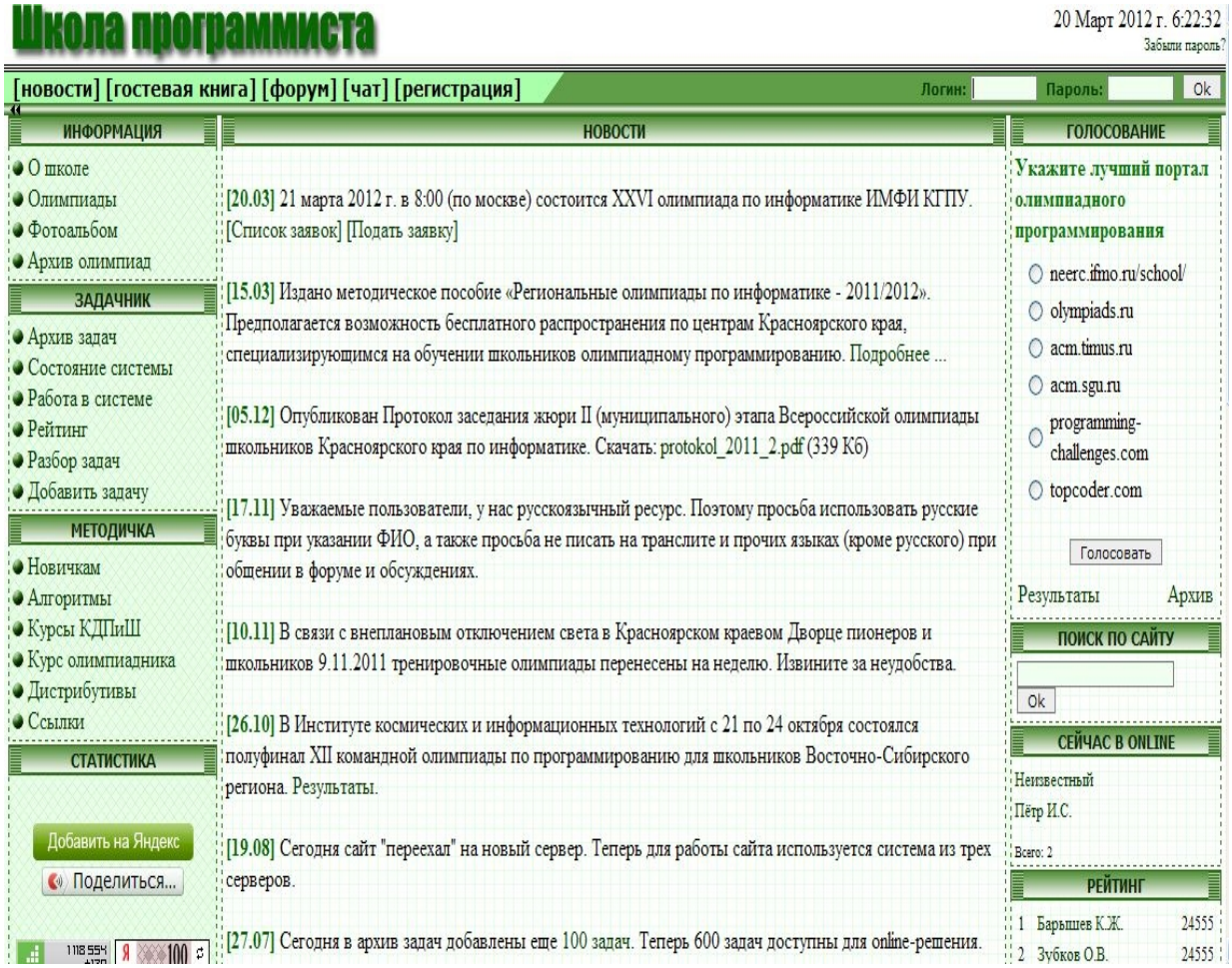
1.1. Dasturlash bo'yicha olimpiadaga oid saytlar.

Asrimiz axborot texnologyalari asri bo`lganligi sabali barcha sohalarda inson mehnatini yengillashtirish maqsadida avtomatlashuv jarayoni kechmoqda. Avtomatlashuv esa usbu soha uchun mo`ljallangan dasturiy ta`minotlarni yaratish orqali amalga oshiriladi. Kompyuterdagi barcha dasturiy taminotlar albatta dasturlash tillari orqali yaratiladi. Biror soha uchun mo`ljallangan dasturiy ta`minotni yaratish uchun o`z navbatida yetuk dasturchi mutaxassilar kerak bo`ladi. Dasturlash bo'yicha qancha ko'p masalalar bajarilsa, dasturchi qobiliyati shu darajada rivojlanadi. Yechilgan masalaning to'g'riligini tekshirish esa ancha vaqtni oladi. Shuning uchun hozirgi kunda internet tarmog'ida bir qancha dasturlash bo`yicha olimpiadaga oid saytlar yaratilgan bo'lib, ularda turli mavzularga doir masalalar berib qo'yilgan. Dasturlashga qiziquvchilar biror masalani yechib, sayt orqali uning to'g'riligini bir necha soniyada tekshirib ko'rishlari mumkin.

Internet tarmog'idagi dasturlash bo'yicha olimpiadaga oid saytlarga quyidagilarni misol keltirish mumkin: acmp.ru, neerc.ifmo.ru, acm.timus.ru, www.e-olimp.com, algolist.manual.ru, acm.albertina.ru hamda o`zbek tilidagi acm.tuit.uz sayti. Bunday saytlarning yaratilishidan maqsad foydalanuvchilarning o'z ustlarida mustaqil shug'ullanishlarini hamda o'z bilimlarini mustahkamlashlarini ta`minlash va ularning dasturlashga bo'lgan qiziqishlarini oshirishdan iborat. Bunda foydalanuvchi biror bir saytga o'zini registrasiya qilib, ya'ni ro'yxatdan o'tib, so'ngra shu saytdagi masalalarni o'zi mustaqil yechish va uni to'g'riligini tekshirish imkoniyatiga ega bo'ladi. Saytlardagi masalalarni Paskal, Delphi, C++, Visual C++ yoki Java dasturlash tillaridan ixtiyoriy birortasida bajarishlari mumkin. Hozirgi kunda talabalar yoki mustaqil shug'ullanuvchilar uchun o'zbek tilida deyarli adabiyotlar mavjud emasligini hisobga olgan holda bitiruv malakaviy ishida o'zbek tilidagi uslubiy qo'llanmani yaratish masalasi qo'yildi. Biror bir dasturlash tilini mukammal o'rgangan va

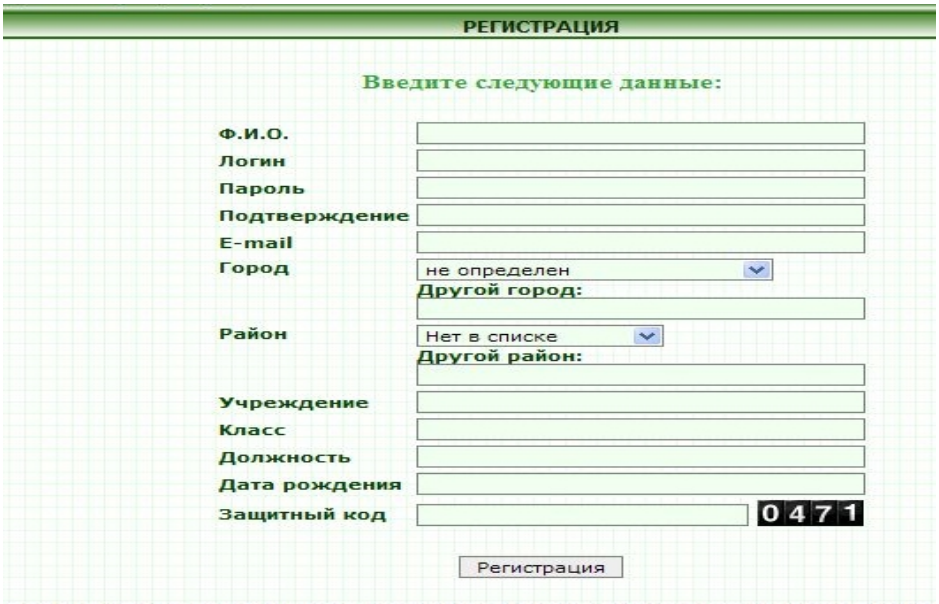
mustaqil o’zi murakkab masalalarni yecha oladigan foydalanuvchilardan kelgusida yetuk dasturchi mutaxasislar yetishib chiqadi.

Bitiruv malakaviy ishda biz acmp.ru sayti to’g’risida ma’lumotlar keltiramiz va Java dasturlash tili yordamida masalalarni bajarib, uni ushbu sayt orqali yuborib tekshirishni qarab chiqamiz. acmp.ru saytiga kiramiz (1- rasm). Quyida saytning ko’rinishi keltirilgan.



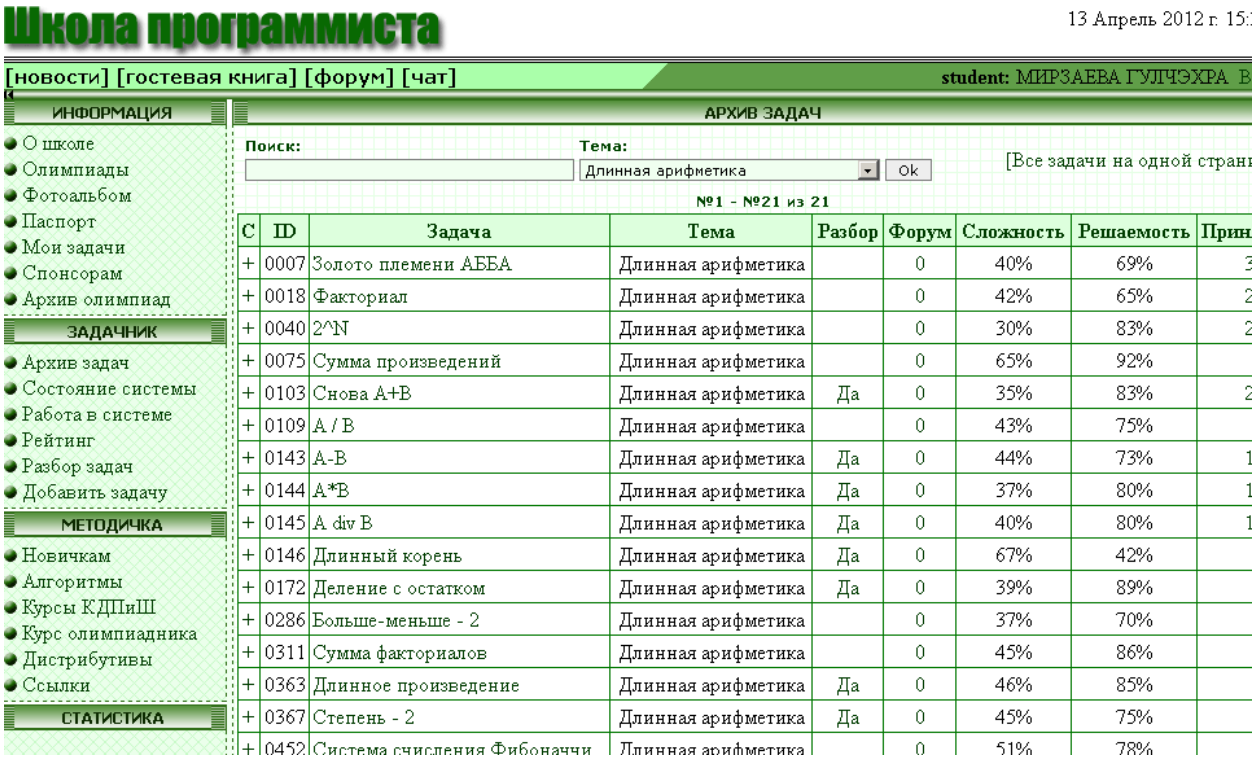
1-rasm. acmp.ru saytini umumiy ko’rinishi.

Hozircha bu saytdan faqat masalalarni ko’rish mumkin. Yechilgan masalalarni tekshirib ko’rish imkoniyati mavjud emas. Buning uchun registrasiya bo’limidan ro`yxatdan o’tib login va parolga ega bo’lamiz. Ro’yxatdan o’tish uchun регистрация bo’limiga kiramiz (2-rasm).



2-rasm. Registrasiya.

Hosil bo`lgan oynadagi bo’sh joylarni to’ldiring va регистрация tugmasini tanlang. Endi siz login va parolga ega bo’ldingiz, istalgan vaqtda bu saytga login va parolingizni kiritib yechilgan masalalarni tekshirib ko`rishingiz mumkin (3-rasm).



3-rasm. Login va parolni kiritib masalalarni ko’rish.

Bu saytda barcha masalalar arxiv zadach bo'limida joylashtirilgan. Arxiv zadach bo'limini tanlasak 3- rasmda keltirilganidek barcha masalalar chiqadi. Foydalanuvchi bu masalalarni mavzu bo'yicha yoki qiyinlik darajasi bo'yicha saralab olishi mumkin. Buning uchun tema bo'limidan ixtiyoriy mavzu tanlanadi, natijada faqat shu mavzuga oid masalalar chiqadi. Agar сложность ustuniga sichqonchani bossak masalalar qiyinlik darajasi bo'yicha saralanadi.

1.2. Java dasturlash tilida olimpiada masalallarini yechish va tekshirish.

Saytdagi barcha masalalarda qiymatlar faylda berilib natija ham faylda chiqariladi. Shuning uchun biz bu bo`limda Java dasturlash tilidan foydalaniladigan operatorlarni va fayllar bilan ishlash imkoniyatlarini batafsil keltiramiz va oxirida biror bir olimpiada masalasini Java dasturlash tilida yechib uni sayt orqali tekshirishni ko'ramiz.

Java dasturlash tilida dastur tuzilishi.

```
class <dastur nomi > {
 public static void main(String args [ ]) {
 <Operatorlar ketma-ketligi>
 }
}
```

Biz hozir Java tilida yozilgan dasturni ko`rsatamiz. Misol uchun Main.java faylini ko`ramiz.

```
class Main {
  public static void main(String args [ ]) {
 System.out.println("Java dasturlash tiliga xush kelibsiz");
  }
}
```

Java dasturlash tilida tiplar.

1. Oddiy tiplar.
2. Murakkab tiplar.

Oddiy tiplar.

1. Butun tiplar.
2. Xaqiqiy tiplar.
3. Simvolli(Xarfiy) tiplar.
4. Mantiqiy tiplar.

Butun tiplar.

Nomi	**Razryadi**	**Chegarasi**
long	**64**	-9, 223, 372, 036, 854, 775, 808 ….. 9, 223, 372, 036, 854, 775, 807
Int	**32**	-2, 147, 483, 648 ….. 2, 147, 483, 647
Short	**16**	-32, 768 …... 32, 767
byte	**8**	-128 ….. 127

Butun tiplar uchun:

1. Ifodadagi barcha operandlar eng kamida int tipiga keltiriladi. Agar ifodada long operand ishtirok etgan bo`lsa, ifodaning tipi long bo`ladi.
2. Agar ifodaning tipi majburiy o`zgartirilsa buning uchun ifodaning qiymati u o`tadigan tip diapazoniga tushishi kerak, aks xolda ifodaning qiymati o`tish kerak bo`lgan tip moduliga bo`linib qoldiq olinadi.
3. Barcha literallar int tipini oladilar. Agar literalning qiymati long diapazoniga o`tsa literalning oxirida l(L) harfi yoziladi.

Haqiqiy tiplar.

Nomi	**Razryadi**	**Chegarasi**
Double	64	1. 7e-308 …. 1. 7e+ 308
Float	32	3. 4e-038 …. 3. 4e+ 038

Harfiy tip(char).

Izoh: **Char** tipi uchun 2 bayt xotira ajratiladi va u ixtiyoriy simvolni saqlaydi. Boshqacha qilib aytganda shu simvolning UNICODE dagi kodini saqlaydi. Ammo bu qiymat bilan ixtiyoriy arifmetik operatsiyastiyalarni bajarish mumkin. Agar

operatsiyaning natijasi yana char tipida saqlansa u UNICODE dagi qandaydir kod deb tasavvur qilinadi. Bu qiymat 0 ÷ 65535 bo`lishi shart.

Mantiqiy tip(Boolean).

Mantiqiy tipning 2 xil qiymati mavjud:

1. false(chin).
2. true(yolg`on).

Murakkab tiplar.

Satr tipi(String).

E'lon qilish:

- String S;
- String S1="Maktab";
- String S2=S1+"N4" ; S2="MaktabN4";

Satr ustida bajariladigan metodlar:

- S2.charAt(1)='a';
- S2.length()=8; s2 satr uzunligi
- S2.indexOf('a')=1; 1-uchragan qavsdagi belgining indeksini qaytaradi.
- S2.indexOf('a',3)=4; 3-belgidan keyingi 'a' ning indeksini qaytaradi
- S2.substring(3)="tabN4"; 3-belgidan boshlab qirqib oladi
- S2.substring(1,5)="aktab"; 1-belgidan 5-sigacha bo`lgan qismni qirqadi
- S1.equals(S2)→false; 2 ta satr tengligini tekshirish

Massiv tipi.

Massiv deb bir necha (soni chegaralangan) bir xil tipga ega bo`lgan elementlarning tartiblashtirilgan joylashuviga aytiladi.

int S[]=new int[10];

S[0]=12; S[3]=35;

long A[][]=new long[4][3];

Arifmetik operatorlar.

Operator	**O'qilishi**

+	Qo`shish
-	Ayirish
*	Ko`paytirish
/	Bo`lish
%	Qoldiqli bo`lish
++	Inkrement

Munosabat operatorlari.

Operator	O'qilishi
==	Teng
!=	Teng emas
>	Katta
<	Kichik
>=	Katta yoki teng
<=	Kichik yoki teng

Mantiqiy operatorlar.

Operator	O'qilishi
&	Mantiqiy va (AND)
|	Mantiqiy yoki (OR)
^	(XOR)
||	Tezkor yoki
&&	Tezkor va
!	Mantiqiy inkor (NOT)
!=	Teng emas
==	Teng

A	V	OR	AND	XOR	NOT A
false	**false**	**false**	**false**	**false**	**true**
true	**false**	**true**	**false**	**true**	**false**
false	**true**	**true**	**false**	**true**	**true**
true	**true**	**true**	**true**	**false**	**false**

Tarmoqlanish operatorlari.

if (< shart >) < operator >

if (< shart >) <operator1 > else < operator2>

switch-case operatori.

```
Switch(<ifoda >){
 case <1-xol >:<1-operatorlar>
 case <2-xol >:<2-operatorlar >
 :
 case <k-xol >:<k-operatorlar >
  default: <operatorlar >
}
```

Ishlash prinsipi: Bu erda **<ifoda> byte, short, int, long** yoki **char** tipiga ega. Ifoda hisoblanib qiymati topiladi va bu qiymat **< 1-hol>,< 2-hol>… < k-hol>** qiymatlar bilan taqqoslanadi.Faraz qilaylik **< i-hol>** qiymatiga teng bo`ldi. U holda **< i- operatorlar>** ketma-ketligidan boshlab ishlaydi. Agar **< ifoda>** ning qiymati biror bir hol bilan mos kelmasa **default** qatoridagi operatorlar ketma-ketligi ishlaydi. Shu bilan **switch-case** operatori uz ishini tugatadi.

Takrorlash operatorlari.

while (< shart >) < operator >

Ishlash prinsipi: Bu yerda **<shart>** mantiqiy ifoda. **<shart>** tekshiriladi agar shart chin natija bersa **<operator >** bajariladi va yana **<shart>** tekshiriladi. Agar **<shart>** yana chin natija bersa, **<operator>** yana bajariladi va yana **<shart>** tekshiriladi va hokazo toki **<shart>** dan yolg`on natija chiqguncha , **<shart>** dan

yolg`on natija chiqdiki **<operator>** bajarilmasdan **while** operatori o`z ishini tugatadi.

Do-while takrorlash operatori.

do <operator> while <shart>

Ishlash prinsipi: Operator bajariladi va undan keyin **<shart>** bajariladi. Agar **<shart>**ning natijasi chin bo`lsa yana **<operator>** bajariladi va yana **<shart>** tekshiriladi va hokazo toki **<shart>**dan yolg`on natija chiqguncha. **<Shart>**dan yolg`on natija chiqdiki **do-while** operatori o`z ishini tugatadi deb hisoblanadi.

Izoh: Bu erda **<operator>** hech bo`lmaganda bir marta ishlaydi.

for ([<initsializatsiya>**];[**<shart>**];[**<orttirma>**]) <operator>**

Ishlash prinsipi: **<initsializatsiya>** qism ishlaydi faqat bir marta undan keyin **<shart>** bajariladi.Agar **<shart>**dan chin natija chiqsa **<operator>** bajariladi va undan keyin **<orttirma>** bajariladi yana **<shart>** tekshiriladi.Agar **<shart>**dan chin natija chiqsa yana **<operator>** bajariladi va undan keyin **<orttirma>** bajariladi yana **<shart>** tekshiriladi. Toki **<shart>**dan yolg`on natija chiqguncha, **<shart>**dan yolg`on natija chiqdiki **for** operatori o`z ishini tugatadi deb hisoblanadi.

For siklining har xil ko`rinishlari.

<initsializatsiya>,<shart>,<orttirma> bo`sh, bitta ifoda, yoki bir nechta ifodadan tashkil topgan bo`lishi mumkin.Agar **<initsializatsiya>** bo`sh bo`lsa **for** operatorining **<initsializatsiya>** qismi ishlamaydi.Agar **<shart>** bo`sh bo`lsa u doimo chin qabul qilinadi. Agar **<orttirma>** bo`sh bo`lsa **for** operatorining **<orttirma>** qismi ishlamaydi.Agar **<operator>** bo`sh bo`lsa u ham ishlamaydi. Agar biror bir qism bir necha ifodadan tashkil topgan bo`lsa ular bir-biridan vergul orqali ajratiladi.

Misol: i dan j gacha sonlar yig`indisini quyidagi ko`rinishlarda yozish mumkin.

1) **S=0;**

   k=i;

   for(;k<=j; S+=k++);

2) **S=0;**
   k=i;
   for(;k<=j; S+=k,k++);
3) **S=0;**
   k=i;
   for(;S+=k,++k<j;);
4) **S=0;**
   k=i;

for(;; S+=k++) if(k>j) break;

5) **S=0;**
   k=i;

for(;;){
if(k>j) break;
S+=k++;
}

break operatori.

1) **break;**
2) **break <metka>;**

<metka>siz **break** 2 maqsadda ishlatiladi.

1) **switch case** operatorida tarmoq ishini tugatish uchun.
2) Siklda sikl ishini tugatish uchun: agar sikllar ichma-ich joylashgan bo`lsa **break** operatori o`zi joylashgan eng ichkaridagi sikl ishini tugatadi.

Metkali **break** operatorini ishlatish uchun shu **break** operator joylashgan biror bir tashqi blok biror bir nom bilan nomlangan bo`lishi kerak. U xolda **break <metka>** operatori bajarilishi natijasida shu **break** operatori joylashgan va **<metka>** nomlangan tashqi blok o`z ishini tugatadi.

a1:{
:
{

if(<shart>) break a1;

: }

: }

continue operatori.

Bu operator **while, do-while, for** operatorlari ichida ishlatiladi.

While(< shart >){

:

continue

:

}

continue operatorining ishlashi natijasida siklning **continue** operatoridan keyingi qismi tashlab yuboriladi, ya`ni **while** va **do-while** sikllarida **<shart>** bajariladi va **<shart>**dan chin natija chiqsa siklning keyingi iteratsiyasi ishlaydi aks holda siklning ishi tugaydi. **for** siklida esa **<orttirma>** ishlaydi va undan keyin **<shart>** bajariladi **<shart>**dan chin natija chiqsa siklning keyingi iteratsiyasi ishlaydi aks xolda sikl ishi tugaydi.

Metkali continue operatori.

Continue < metka >;

Ichma-ich joylashgan sikllar holatida agar **continue** operatori ichkaridagi siklning ichida joylashgan bo`lsa va u **<metka>**siz bo`lsa u shu ichkaridagi siklni keyingi iteratsiyasini boshlaydi.

Agar tashqi siklga **<metka>** qo`yilgan bo`lsa ichki siklda joylashgan shu **<metka>**li **continue** operatori tashqi siklning keyingi iteratsiyasini boshlaydi.

continue operatori ishlashi natijasida **for** siklining keyingi iteratsiyasi boshlanadi.

return operatori.

return operatori ishlashi natijasida ishlayotgan programma(metod) darhol o`z ishini tugatadi va boshqarish joriy programmani(metodni) qaysi programma

(metod) chaqirgan bo`lsa o`shanga uzatiladi. Agar joriy programma **main** metodi bo`lsa boshqarish operastion tizimga uzatiladi.

Uzun sonlar bilan ishlash.

BigInteger va BigDecimal sinflari

Butun sonlar uchun: BigInteger sinfi.

Natija tipi	**Amal nomi(qiymat tipi)**	**Izoh**
BigInteger	abs()	Sonnig moduli hisoblaydi
BigInteger	add(BigInteger val)	BigInteger tipidagi 2 sonni qo`shish
BigInteger	and(BigInteger val)	
int	compareTo(BigInteger val)	BgInteger tipida ikki sonni solishtirish. a.compareTo(b): agar a>b bo`lsa 1; aks holda -1 natijani qaytaradi.
BigInteger	divide(BigInteger val)	Ikki sonni bo`lish: ”a:b”→a.divide(b)
BigInteger	max(BigInteger val)	Ikki sondan kattasini aniqlaydi
BigInteger	min(BigInteger val)	Ikki sondan kichigini aniqlaydi
BigInteger	mod(BigInteger val)	bo`linmaning butun qismi
BigInteger	multiply(BigInteger val)	Ikki sonni ko`paytirish: A*B→A.multiply(B)
BigInteger	not()	Inkor amali
BigInteger	or(BigInteger val)	Yoki mantiqiy amali
BigInteger	pow(int exponent)	Daraja
BigInteger	Subtract(BigInteger val)	Ayirish: A-B→A.substract(B)
String	toString()	Sonni satrga aylantiradi
String	toString(radis)	Sonnig biror sanoq tizimidagi ko`rinishini satrga aylantiradi.Bu

yerda radis sanoq tizimi.

Haqiqiy sonlar uchun:BigDecimal sinfi.

BigDecimal sinfida ham BigInteger sinfidagi barcha amallar qo`llaniladi.

Qo`shimcha:

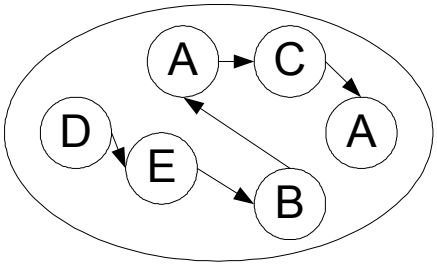
BigDecimal round(MathContext me)

Kolleksiyalar

- Kolleksiya ─ tartiblanmagan elementlar to`plami
- Collection Interfeysi

Iteratorlar

- Iterator ─ kolleksiya bo`ylab o`tish
- Iterator Interfeysi
- Iterator Collection.iterator() metodi

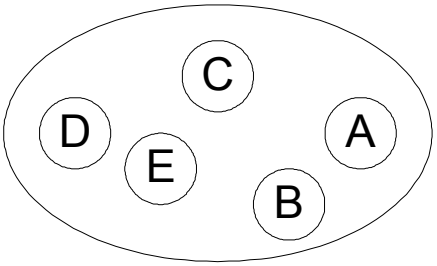


Iteratorlar metodi

- hasNext() ─ keyingi elementi borligini aniqlash
- next() ─ keyingi elementini olish
- remove() ─ elementini o`chirish
- Favqulotda
  - NoSuchElementException ─ kolleksiya oxiriga kelgandan keyingi xatolik
  - ConcurrentModificationException ─ kolleksiyani o`zgartirish vaqtidagi xatolik

To`plam

- To`plamlar ─ Takrorlanmaydigan elementlardan tashkil topgan kolleksiya
- Set Interfeysi



Elemenlarni taqqoslash

- Metod Object.equals(Object object)
- Refleksivlik

o1.equals(o1)

- Simmetrlilik

o1.equals(o2) == o2.equals(o1)

- Tranzitivlik

o1.equals(o2) && o2.equals(o3) => o1.equals(o3)

- Barqarorlik

Agar o1va o2 o`zgarmasa o1.equals(o2) ning natijasi o`zgarmaydi

- Null qiymatni qayta ishlash.

o1.equals(null) == false

To`plamlar ustida amallar

- addAll(Collection c) – To`plamlarni birlashmasi
- retainAll(Collection c) – To`plamlarni kesishmasi
- containsAll(Collection c) – s to`plam mavjudligini tekshiradi
- removeAll(Collection c) – To`plam ayirmasi

HashSet va LinkedHashSet sinflari

- HashSet ─ xesh asosida to’plam
- LinkedHashSet ─ xesh asosida to’plam tartibi saqlangan holda

HashSet Konstruktorlari

- HashSet() ─ bo’sh to’plami
- HashSet(Collection c) ─ kolleksiya elementlari
- HashSet(int initialCapacity) ─ Boshlang`ich sig`imi (vmestimost)
- HashSet(int initialCapacity, double loadFactor) ─ Boshlangich sig`imi (vmestimost) va to’ldirish darajasi

Ro’yxat bo’yicha iterator amallari

- Harakatlanish
  - hasNext() / hasPrevious() ─ tekshirish

  - next() / previous() ─ elementni olish
  - nextIndex() / previousIndex() ─ indeksini aniqlash
- O’zgartirish
  - remove() ─ elementni o’chirish
  - set(Object e) ─ elementni o’zgartirish
  - add(Object e) ─ elementni qo’shish

ArrayList sinfi

- ArrayList ─ massiv asosida ro’yxat
- Yutug`i
  - Indeksi bo’yicha tez murojaat
  - Oxiridan elementlarni qo’shish yoki o’chirish tez bajariladi
- Kamchiligi
  - Elementlarni qo’shish yoki o’chirish sekin bajariladi

ArrayList konstruktorlari

- ArrayList() ─ bo’sh ro’yxat
- ArrayList(Collection c) ─ kolleksiyani nusxalash
- ArrayList(int initialCapacity) ─ berilgan sig`imli bo’sh ro’yxat

ArrayList dan foydalanish

- “Cheksiz” massiv (Chegaralanmagan)
- Stek

Misol. ArrayList ni ekran chiqarish

```
List list = new ArrayList();
…
for (int i = list.size() - 1; i >= 0; i--) {
System.out.println(list.get(i));          }
```

Dasturga izoh yozish

Birta satrdan iborat izoh yozish uchun ─

// …

belgi qo`llanadi.

Bir nechta satrlardan iborat izoh yozish uchun esa

/*

…

*/

belgidan foydalaniladi.

Ma`lumotlarni fayldan o`qib olish va faylga yozish

Ma`lumotlarni fayldan o`qib olish uchun avvalambor qaralayotgan sinfga util paketini import qilish kerak. Util paketining File sinfi orqali tashqi faylni bog`lash, Scanner sinfi orqali ma`lumotlarni bog`langan fayldan o`qib olish, PrintWriter sinfi orqali esa ma`lumotlarni bog`langan faylga chiqarish mumkin.

Scanner sinfining tashqi faylga bog`langan obyektini yaratish:

Scanner <obyekt nomi>=new Scanner(new File(<tashqi fayl nomi>));

Bunda obyekt nomi-ixtiyoriy nom, tashqi fayl nomi-kengaytmasi .txt bo`lgan fayl.

Scanner sinfi obyektlarining quyidagi metodlari mavjud:

- next()-fayldagi ma`lumotlarni satr ko`rinishida(bo`sh joygacha) o`qib olish. Bu metodning quyidagi ko`rinishlari ham mavjud: nextInt(), nextDouble(), nextLong()-ma`lumotlarni nextdan keyin kelgan tipda o`qib oladi.
- nextLine()-faydan ma`lumotlarni satr ko`rinishida(to`liq bir satr) o`qib oladi.
- hasNext()- faylda ma`lumotlar mavjud yoki yo`qligini tekshiradi, natijasi true yoki false bo`ladi. Natija true qiymat bo`lsa fayl oxiriga borilmaganini, aks holda faylda boshqa ma`lumotlar yo`qligini bildiradi.

PrintWriter sinfining tashqi faylga bog`langan obyektini yaratish:

PrintWriter <obyekt nomi>=new PrintWriter (new File(<tashqi fayl nomi>));

PrintWriter sinfi orqali ma`lumotlarni bog`langan faylga chiqarish uchun quyidagi metodlardan foydalaniladi:

- print()-ma`lumotlarni bog`langan faylga ketma-ket chop etadi.
- println()-ma`lumotlarni bog`langan faylga alohida satrlarda chop etadi.

Java dasturlash tilida olimpiada masalalarini yechish

Quyida acmp.ru saytidagi Oddiy matematika mavzusidan olingan “Arifmetik progressiya” masalasining java dasturlash tilidagi dasturini tuzishni ko`rib chiqamiz. Buning uchun masala sharti bilan tanishib chiqamiz. Masala shartini esa quyidagi oynadagi zadacha ustunidan masala nomini tanlash orqali ko`rish mumkin:

АРХИВ ЗАДАЧ

Поиск: Тема: Простая математика Ok [Все задачи на одной странице]

№1 - №25 из 25

C	ID	Задача	Тема	Разбор	Форум	Сложность	Решаемость	Принято
	0312	Арифметическая прогрессия	Простая математика	Да	0	15%	97%	4460
	0086	Головоломка про ферзей	Простая математика		0	15%	95%	3312
	0062	Клетки	Простая математика		0	15%	96%	3940
	0010	Уравнение	Простая математика		0	17%	94%	4187
	0063	Загадка	Простая математика	Да	0	18%	93%	4045
	0340	Коробки	Простая математика		0	19%	80%	1667
	0276	Разбиение на части	Простая математика		0	21%	90%	1807
	0439	Закон Амдала	Простая математика		0	23%	94%	1115
	0520	Оптовая покупка	Простая математика		0	27%	56%	362

4-rasm.Tanalngan mavzuga doir masalalar ro`yxati.

Natijada masala sharti yozilgan oyna hosil bo`ladi:

ЗАДАЧА №312

Арифметическая прогрессия

(Время: 1 сек. Память: 16 Мб Сложность: 15%)

Заданы первый и второй элементы арифметической прогрессии. Требуется написать программу, которая вычислит элемент прогрессии по ее номеру.

Входные данные

Входной файл INPUT.TXT содержит три целых числа, разделенных пробелами – первый элемент прогрессии A_1 ($1 \le A_1 \le 1000$), второй элемент прогрессии A_2 ($1 \le A_2 \le 1000$), и номер требуемого элемента N ($1 \le N \le 1000$).

Выходные данные

Выходной файл OUTPUT.TXT должен содержать одно целое число - N-й элемент арифметической прогрессии.

Пример

№	INPUT.TXT	OUTPUT.TXT
1	1 5 3	9

5-rasm.Tanlangan masala sharti.

Ushbu masalaning java dasturlash tilidagi dasturi quyidagicha :

```
import java.util.*; import java.io.*;
/*util-fayllar bilan ishlashga imkon beruvchi paket, io-xatoliklar to`g`risidagi foydalanuvchiga malumot berilishini ta`minlovchi paket*/
class Main{
//Java dasturlash tilining asosiy elementi sinfni e`lon qilish
 public static void main(String mm[])
/*main metodi joylashgan sinf asosiy sinf hisoblanadi, public-main metodini ishlatishga ruxsat beradi, static main metodining statik metod enanligini, void ushbu metodning hech qanday natija qaytarmasligini bildiradi, String mm[] kiritiladigan ma`lumotlarni satr tipli massivga joylashtiradi va ulardan foydalanish imkoniyati mavjud bo`ldadi. */
throws IOException {
//xatoliklar to`g`risida ma`lumot beradi, masalan input.txt fayli mavjud bo`lmasa
Scanner sc=new Scanner(new File("input.txt"));
/*util paketining ma`lumotlarni o`qib olinishini taminlovchi Scanner sinfining obyektini yasash.Obyekt nomi sc, u input.txt fayli bilan bog`langan*/
PrintWriter pw=new PrintWriter(new File("output.txt"));
/*util paketining ma`lumotlarni chiqarilishini taminlovchi PrintWriter sinfining obyektini yasash.Obyekt nomi pw, u output.txt fayli bilan bog`langan*/
int a1=sc.nextInt(),a2=sc.nextInt(),n=sc.nextInt();
/*a1-arifmetik progressiyaning birinchi hadi, a2-arifmetik progressiyaning ikkinchi hadi, n-arifmetik progressiyada topilishi kerak bo`lgan element nomeri*/
int d=a2-a1; // Arifmatik progressiya ayirmasi
int an=a1+d*(n-1);  //Arifmetik progressiyaning n- hadini topish
pw.print(an); //Topilgan elementni output.txt fayliga chiqarish
pw.close(); } } //output.txt faylini yopish
```

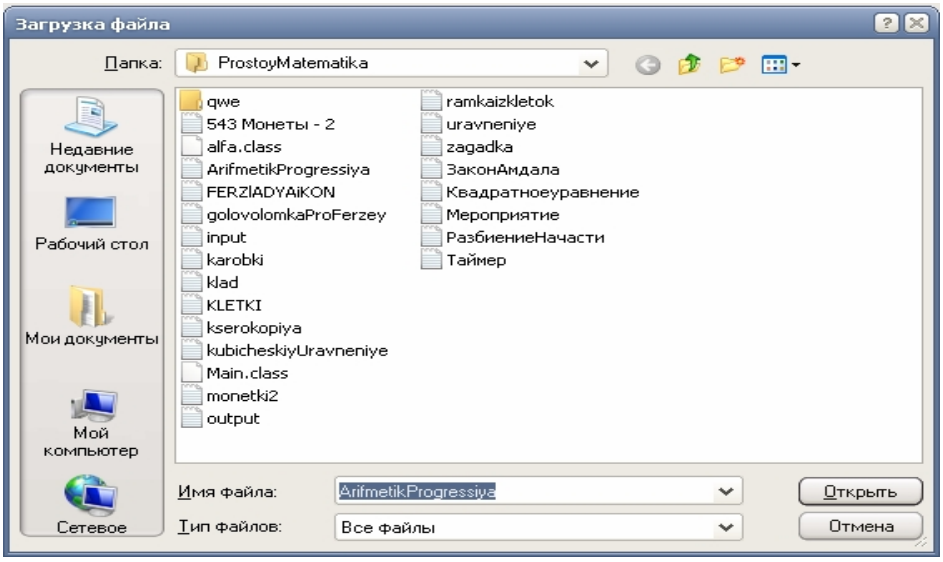
Dasturni acmp.ru sayti orqali tekshirish

Dasturni acmp.ru sayti orqali tekshirib ko`rish uchun(bunda foydalanuvchi ushbu saytda registratsiya qilingan bo`lishi kerak) masala sharti bayon qilingan

oyna so`ngida ”Masalani yuborish(сдать задачу)” bandidagi “обзор” tugmasini bosish;

6-rasm. Masalani yuborish(сдать задачу) bandi ko`rinishi.

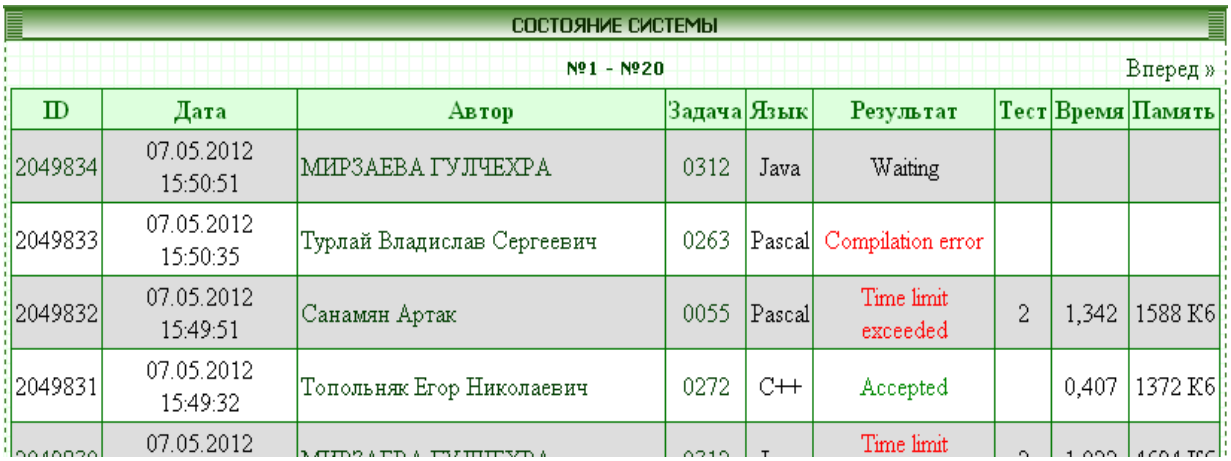
Hosil bo`lgan oynada masala yechimi mavjud bo`lgan fayl yo`lini ko`rsatish va Открыть tugmasini bosish;



7-rasm. Masala yechimi mavjud bo`lgan fayl yo`lini ko`rsatish.

Undan so`ng “Jo`natish” Отправить tugmasini bosish kerak.

Natijada quyidagi oyna hosil bo`ladi:



СОСТОЯНИЕ СИСТЕМЫ

№1 - №20 Вперед »

ID	Дата	Автор	Задача	Язык	Результат	Тест	Время	Память
2049834	07.05.2012 15:50:51	МИРЗАЕВА ГУЛЧЕХРА	0312	Java	Waiting			
2049833	07.05.2012 15:50:35	Турлай Владислав Сергеевич	0263	Pascal	Compilation error			
2049832	07.05.2012 15:49:51	Санамян Артак	0055	Pascal	Time limit exceeded	2	1,342	1588 K6
2049831	07.05.2012 15:49:32	Топольняк Егор Николаевич	0272	C++	Accepted		0,407	1372 K6
[illegible]	07.05.2012	[illegible]	[illegible]	[illegible]	Time limit	[illegible]	[illegible]	[illegible]

8-rasm. Masala yuborilgach hosil bo`luvchi ko`rinish.

Agar yuqoridagi oynaning Результат(natija) bandida:

- Accepted- javobi bo`lsa , u holda ushbu masala to`g`ri yechilgan bo`ladi;
- Waiting- javobi bo`lsa, F5 yoki [Ctrl+r] tugmalaridan birini bosish orqali oynani yangilash lozim bo`ladi.
- Compiling -javobi esa yuborilgan masalaning kompilyatsiya qilinayotganligini bildiradi.Bu holda ham F5 yoki [Ctrl+r] tugmalaridan birini bosish orqali oynani qayta yuklash lozim bo`ladi.
- Running-yuborilgan dasturni testlash jarayoni olib borilayotganligini bildiradi. Dastur test bandida ko`rsatilgan testni bajarayotgan bo`ladi.
- Wrong answer - noto`g`ri javob. Dastur natijasi masala yechimi bo`la olmaydi ya`ni natija noto`g`ri formatda chiqarilgan yoki dastur algoritmi noto`g`ri tuzilgan. Dastur berilgan testni bajarishda noto`gri natija qaytargan.
- Time limit exceeded-dastur masala yechimini ko’rsatilgan vaqtdan ortiq muddatda chiqarayotganligini bildiradi. Bunda masala yechimiga olib boruvchi dastur effiktiv bo`lmasligi yoki algoritmik xatolar mavjud bo`lishi mumkin.
- Presentation Error-natija chiqariluvchi Output.txt fayli mavjud emasligini bildiradi yoki ushbu fayl boshqa nom bilan hosil qilingan bo`lishi mumkin.
- Compilation error-komplyatsiya xatoligi. Komplyatsya jarayonida dasturda sintaktik xatoliklar aniqlangan bo`lishi yoki ko`rsatilgan fayl noto`g`ri kengaytmaga ega bo`lishi mumkin.
- Memory limit exceeded- Dasturni bajarishga belgilanganidan ortiq xotira sarflanayotganligini yoki dastur algoritmining effektiv emasligini bildiradi.
- Runtime error- dastur bajarilishi vaqtidagi xatolik. Masalan dastur algoritmida: nolga bo`lish, massivning mavjud bo`lmagan elementiga murojat va h.k xatoliklar mavjud bo`lsa. C++ dasturlash tilida bu holatni "return 0" operatorini yozish orqali bartaraf qilish mumkin.

Agar yuborilgan masala barcha testlardan o`tgan bo`lsa, foydalanuvchining Результат(natija) bandida “Accepted” javobi bo`ladi.

Endi masalalar arividan yechilgan masala mavzusiga kirib, hosil bo`lgan jadvalning C ustunida bu masalaning to`g`riligini tasdiqlovchi “+” belgisi hosil bo`lganini ko`rish mumkin.

C	ID	Задача	Тема	Разбор	Форум	Сложность	Решаемость	Принято
+	0312	Арифметическая прогрессия	Простая математика	Да	0	15%	97%	4461
+	0086	Головоломка про ферзей	Простая математика		0	15%	95%	3312
+	0062	Клетки	Простая математика		0	15%	96%	3941
+	0010	Уравнение	Простая математика		0	17%	94%	4190
+	0063	Загадка	Простая математика	Да	0	18%	93%	4048
+	0340	Коробки	Простая математика		0	19%	80%	1667
+	0276	Разбиение на части	Простая математика		0	21%	90%	1808

9-rasm.Mavzuga doir yechilgan masalalar ro`yxati.

“+” belgisi chertish orqali yuborilgan dastur kodini olish mumkin.

I bobning qisqacha xulosasi.

I bob internet tizimida olimpiada masalalari va ularni yechishga bag’ishlangan bo`lib, 2 ta bo`limdan iborat.

1-bo`lim “Dasturlash bo’yicha olimpiadaga oid saytlar” deb nomlanib, internet tarmog’idagi dasturlash bo’yicha olimpiadaga oid turli xil saytlar, hamda ushbu saytlardan acmp.ru sayti va saytdagi masalalarni ko`rish, yozilgan dasturlarning to`g`riligini tekshirish imkoniga ega bo`lish uchun bajarilishi kerak bo`lgan ishlar ketma-ketligi keltirildi.

2-bo`lim “Java dasturlash tilida olimpiada masalallarini yechish va tekshirish” deb nomlanadi. Bu bo`limda olimpiada masalalarini yechish uchun java dasturlash tilidan foydalaniladigan operatorlar, acmp.ru saytidagi biror masala shartini ko`rish, ushbu masala uchun yozilgan dasturni sayt orqali tekshirish va tekshirish natijasi to`g`risidagi ma`lumotlar keltirildi.

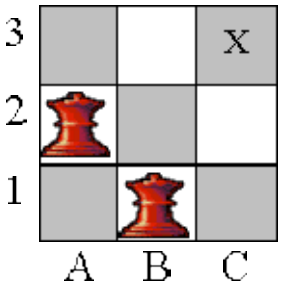
II-BOB. Mavzulashtirilgan olimpiada masalalari

2.1. Oddiy matematika(Простая математика)

№86 Masala

Farzin boshqotirmasi

Vali shaxmat bilan shug'ullana boshladi va bitta boshqotirmani o'qidi. Unda shaxmat doskasiga farzinlarni maksimal sonda shunday qo'yish kerakki, hech bo'lmaganda 1 ta katakka farzin yura olmasin. 3x3 doska uchun bu oson hal qilinadi. NxN doska uchun hal eting.



Kiritiluvchi ma`lumotlar: INPUT.TXT faylida N (1<=N<=100) doska o'lchami berilgan.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida shartga javob beruvchi farzinlarning maksimal sonini chiqaring.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	3	2

Qo`yilgan masalaning dasturi Java dasturlash tilida quyidagicha bo`ladi:

```
import java.util.*; import java.io.*;
class Main{ public static void main(String mm[])throws IOException{
Scanner sc=new Scanner(new File("input.txt"));
PrintWriter pw=new PrintWriter(new File("output.txt"));
int n=sc.nextInt(),s; //n- doska o`lchami, s- farzinlarning maksimal soni
if (n>2) s=n*n-(3*n-2); else s=0;
pw.print(s); // s ni output.txt fayliga chop qilish
pw.close(); } }
```

Keyingi masalalarning dasturini faqat asosiy qismini keltiramiz, ya`ni barcha masalalarda dastur tuzilishi quyidagicha bo`ladi. Faqat asosiy qism o`zgaradi.

```
import java.util.*; // fayllar bilan ishlashga imkon beruvchi paket
import java.io.*; // xatoliklar to`g`risida foydalanuvchiga ma`lumot beruvchi paket
```

class Main{//asosiy sinf nomi Main bo`lishi kerak

public static void main(String mm[])

/* main metodi joylashgan sinf asosiy sinf hisoblanadi, public-main metodini ishlatishga ruxsat beradi, static-main metodining static ekanligini anglatadi, void-main metodining natija qaytarmasligini ifodalaydi*/

throws IOException{

/*dasturdagi xatoliklar to`g`risida foydalanuvchiga ma`lumot berish, masalan input,output fayllarning mavjud emasligi, nolga bo`lish, massiv chegarasidan oshish*/

Scanner sc=new Scanner(new File("input.txt"));

/*ma`lumotlarni input faylidan o`qib olish uchun Scanner sinfining sc obyektini yaratish */

PrintWriter pw=new PrintWriter(new File("output.txt"));

/*ma`lumotlarni output faylidan o`qib olish uchun PrintWriter sinfining pw obyektini yaratish */

Asosiy qism

pw.close();

// yozilgan ma`lumotlar o`chib ketmasligi uchun output.txt faylini yopish

} }

№340 Masala

Qutilar

A1 × B1 × C1 va A2 × B2 × C2 o’lchamli qutilar berilgan. Ulardan birini ikkinchisining ichiga joylash mumkinmi-yo’qmi aniqlang – bunda qutilarning ixtiyoriy qirrasi atrofida 90 gradusga burish mumkin.

Kiritiluvchi ma`lumotlar: INPUT.TXT da 1- qatorga A1, B1 va C1, 2-qatorga esa A2, B2 va C2 sonlar berilgan. Barcha sonlar musbat va 100 dan katta emas.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida agar

- qutilar bir xil bo’lsa, “Qutilar teng” deb;

- 1-sini 2-qutining ichiga solish mumkin bo'lsa, "1- quti 2-sidan kichikroq" deb;

- 2- quti 1-sining ichiga joylashishi mumkin bo'lsa, "1-quti 2-sidan kattaroq" deb;

Aks holda "bir-biriga joylashtirib bo'lmaydi" deb chop eting.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	1 2 3 3 2 1	Boxes are equal
2	2 2 3 3 2 1	The first box is larger than the second one
3	2 2 3 3 2 3	The first box is smaller than the second one
4	3 4 5 2 4 6	Boxes are incomparable

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int a1[]={sc.nextInt(),sc.nextInt(),sc.nextInt()};
//birinchi quti o`lchamlaridan iborat massiv
int a2[]={sc.nextInt(),sc.nextInt(),sc.nextInt()};
//ikkinchi quti o`lchamlaridan iborat massiv
int c,i,j; boolean k=false;
for(i=0;i<3;i++) { for(j=0;j<2;j++)
if(a1[j]>a1[j+1]) {c=a1[j];a1[j]=a1[j+1];a1[j+1]=c;} } //1-massivni tartiblash
for(i=0;i<3;i++) { for(j=0;j<2;j++)
if(a2[j]>a2[j+1]) {c=a2[j];a2[j]=a2[j+1];a2[j+1]=c;} } //2-massivni tartiblash
if((a1[0]==a2[0])&(a1[1]==a2[1])&(a1[2]==a2[2])){
pw.println("Boxesare equal"); k=true; } //qutilar tengligini tekshirish
if ((a1[0]<=a2[0])&(a1[1]<=a2[1])&(a1[2]<=a2[2])) {
```

```
if (!k) pw.println("The first box is smaller than the second one");
k=true; } //1- quti 2-sidan kichikroq ekanini tekshirish
if ((a1[0]>=a2[0])&(a1[1]>=a2[1])&(a1[2]>=a2[2])) {
if (!k) pw.println("The first box is larger than the second one");
k=true; } //1-quti 2-sidan kattaroq ekanini tekshirish
if (!k) pw.println("Boxes are incomparable");
// qutilarni bir-biriga joylashtirib bo'lmaydigan holat
```

№411 Masala

Kvadrat tenglama

$ax^2 + bx + c = 0$ kvadrat tenglamani a, b, c koeffitsientlar berilgan holda yechish dasturini tuzing.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylida 3 ta son a,b va c berilgan, har biri modul bo'yicha 30000 dan katta emas.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida tenglama ildizlari sonini chiqaring. So'ng alohida satrlarda ildizlarning o'zini 10^{-4} dan katta bo'lmagan xatolik bilan chop eting. Agar ildizlar cheksiz ko'p bo'lsa, yagona satrda -1 ni chop eting.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	1 -3 2	2 1.000000 2.000000
2	0 -1 6	1 6.000000

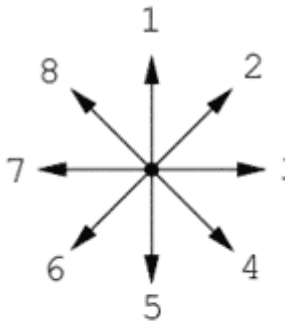
Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
double a=sc.nextDouble(),b=sc.nextDouble(),c=sc.nextDouble();
// a,b,c koeffitsientlarni o`qib olish
double x=0,y=0,d; String s=""; d=b*b-4*a*c; int j=-1,i,n;
```

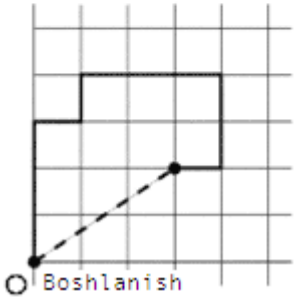
```
if((a==0)&&(b==0)&&(c==0)) j=0;
//agar a,b,c larning har biri nolga teng bo`lsa ildizlar soni j ning nolga tengligi
else if((a==0)&&(b==0)&&(c!=0)) pw.println("0");
//agar a,b nolga teng, c nolga teng emas bo`lgan hol
else if((a==0)&&(b!=0)) { j=1; x=c*(-1.0)/b;} //b*x+c=0 ko`rinishdagi tenglama
else if(d>0){ //diskreminant noldan katta bo`lgan hol
j=2; x=(-1*b-Math.sqrt(b*b-4*a*c))*1.0/(2*a);
y=(-1*b+Math.sqrt(b*b-4*a*c))*1.0/(2*a); }
else if(d==0){ // diskreminant nolga teng bo`lgan hol
j=1; x=(-1*b-Math.sqrt(b*b-4*a*c))*1.0/(2*a); }
if(j>0) { //1 ta ildiz mavjud bo`lgan hol
pw.println(j); pw.printf("%.6g",x); if(j>1) { //2 ta ildiz mavjud bo`lgan hol
pw.println(); pw.printf("%.6g",y); } }
if(d<0) pw.println("0"); // diskreminant noldan kichik bo`lgan hol
if(j==0) pw.println("-1"); //cheksiz ko`p ildizlar mavjud bo`lsa
```

№207 Masala

Xazina



Yashirilgan xazinani topish uchun xaritadan foydalaniladi, xaritada qaysi yo’nalishda necha qadam yurish kerakligi ko’rsatilgan. Yo’nalishlar raqamlar bilan ifodalanadi: 1- shimol, 2 –shimoli-sharq, 3-sharq, 4- janubi-sharq, 5- janub, 6- janubi-g’arb, 7- g’arb, 8- shimoli-g’arb. Har bir yo’nalish uchun 1 ta qadam uzunligi 1 ga teng.



Kiritiluvchi ma`lumotlar: INPUT.TXT faylida 1-satrda N – buyruqlar soni (1≤N≤40). Keyingi N ta qatorda yo’nalish nomeri (1 dan 8 gacha butun son) va qadamlar soni (1 dan 1000gacha butun son) berilgan.

**Chiqariluvchi ma`lumotlar: OUTPUT.TXT** faylida X va Y koordinatalarni chiqaring(2 ta haqiqiy son, bo`sh joy bilan ajratilgan). OX o’qi sharqqa, OY o’qi

shimolga yo'nalgan. Izlash koordinatalar boshidan boshlanadi. Koordinatalarni 10^{-3} aniqlik bilan chiqaring.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	6 1 3 3 1 1 1 3 3 5 2 7 1	3.000 2.000
2	1 8 10	-7.071 7.071

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int a,b,n,i; n=sc.nextInt(); double d=0,x=0,y=0;
for(i=0;i<n;i++) { a=sc.nextInt(); //yo`nalish nomeri
b=sc.nextInt();//qadamlar soni
// yo`nalish nomeriga qarab harakatdan keyingi koordinatani aniqlash→
switch (a) { case 1: y=y+b; break;
case 2: d=b/Math.sqrt(2); x=x+d; y=y+d; break; case 3: x=x+b; break;
case 4: d=b/Math.sqrt(2); x=x+d; y=y-d; break; case 5: y=y-b; break;
case 6: d=b/Math.sqrt(2); x=x-d; y=y-d; break; case 7: x=x-b; break;
case 8: d=b/Math.sqrt(2); x=x-d; y=y+d; break; } }
pw.print(x+" "+y);
```

№205 Masala

Vaqt hisoblagich

Vaqt hisoblagich orqali ma'lum vaqt intervalidan so'ng signal chalinishi lozim bo`lgan vaqtni aniqlang.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylida 1-satrda joriy vaqt ЧЧ:MM:CC formatda (oldida nollari bilan) berilgan. Ushbu cheklovlar o'rinli: ЧЧ - 00 dan 23 gacha, MM va CC - 00 dan 60 gacha.

2-satrda o'lchanadigan vaqt intervali beriladi. Interval Ч:M:C formatda (bu yerda Ч, M va C - 0 dan 10^9 gacha, oldida nollarsiz) yoziladi. Yana agar Ч=0 (yoki Ч=0 va M=0) bo`lsa, ularni yozmaslik mumkin. Masalan: 100:60 bu 100 minut 60 sekund, yoki 101:0, yoxud 1:41:0 degani. 42 esa 42 sekundni bildiradi. 100:100:100 esa - 100 soat, 100 minut, 100 sekund, yoki 101:41:40 deganidir.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida signal chalinadigan vaqtni ЧЧ:MM:CC formatda chiqaring. Bunda agar signal shu sutkada chalinmasa, + days yozuvini qo'shish kerak. Masalan: signal ertasi kuni chalinsa, +1 days yozuvi qo'shiladi.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	01:01:01 48:0:0	01:01:01+2 days
2	01:01:01 58:119	02:01:00
3	23:59:59 1	00:00:00+1 days

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
String s1=sc.next(),s2=sc.next(); //s1-joriy vaqt, s2- vaqt intervali
int soat1=0,soat2=0,sekund1=0,sekund2=0,minut1=0,minut2=0,i,n,k=0,z=0;
if(s1.charAt((short)0)=='0') soat1=Integer.parseInt(s1.substring(1,2));
else soat1=Integer.parseInt(s1.substring(0,2)); //joriy soat
if(s1.charAt((short)3)=='0') minut1=Integer.parseInt(s1.substring(4,5));
else minut1=Integer.parseInt(s1.substring(3,5)); //joriy minut
if(s1.charAt((short)6)=='0') sekund1=s1.charAt(7)-'0';
```

```
else sekund1=Integer.parseInt(s1.substring(6,7)+s1.charAt(7)); //joriy sekund
if(s2.indexOf(':')==-1) sekund2=Integer.parseInt(s2);
//vaqt intervali faqat sekunddan iborat bo`lgan hol
n=s2.length(); int a[]=new int[2];
for(i=0;i<n;i++) if (s2.charAt(i)==':') { a[k]=i; k++; }
if(k==1) { a[1]=0; minut2=Integer.parseInt(s2.substring(0,a[0]));
sekund2=Integer.parseInt(s2.substring(a[0]+1,n)); }
//vaqt intervali sekund va minutdan iborat bo`lgan hol
else if(k!=0) { soat2=Integer.parseInt(s2.substring(0,a[0]));
minut2=Integer.parseInt(s2.substring(a[0]+1,a[1]));
sekund2=Integer.parseInt(s2.substring(a[1]+1,n)); }
//vaqt intervali soat, sekund va minutdan iborat bo`lgan hol
int sekund=(sekund1+sekund2)%60; //ma`lum vaqt o`tgandan keyingi sekund
if((sekund1+sekund2)/60>0) z=(sekund1+sekund2)/60;
int minut=(minut1+minut2+z)%60; // ma`lum vaqt o`tgandan keyingi minut
if((minut1+minut2+z)/60>0) z=(minut1+minut2+z)/60;
int soat=(soat1+soat2+z)%24; // ma`lum vaqt o`tgandan keyingi soat
if((soat1+soat2+z)/24>0) z=(int)(soat1+soat2+z)/24;else z=0;
String s="";
if(sekund>9) s=":"+sekund+s; else s=":"+"0"+sekund+s;
if(minut>9) s=":"+minut+s; else s=":"+"0"+minut+s;
if(soat>9) s=soat+s; else s="0"+soat+s; if(z>0) pw.print(s+"+"+z+" days");
 else pw.print(s);
```

№543 Masala

Tangalar-2

N ta savatda oltin tangalar mavjud. Savatlar 1 dan N gacha bolgan sonlar bilan nomerlangan. Birta savatdan tashqari barcha savatlardagi tangalarning har biri w grammdan og`irlikga ega. Faqat 1 ta savatda qalbaki tangalar joylashgan va uning og`irligi w–d grammni tashkil etadi. Sehirgar birinchi savatdan 1 ta tanga, ikkinchi

savatdan 2 ta tanga va h.k. oxirida (N-1) – savatdan (N-1) ta tanga oldi. N-savatdan esa hech narsa olmadi. U olingan barcha tangalarning massasini hisobladi va shu zaxotiyoq qalbaki tanga joylashgan savatni ko`rsatdi.

Ushbu sehrgarlikni amalga oshiruvchi dastur tuzing.

**Kiritiluvchi ma`lumotlar:INPUT.TXT** faylida 4 ta butun son berilgan: N, w, d va P – ajratib olingan tangalarning massalari yig`indisi ($2 \le N \le 8000$, $1 \le d < w \le 30$).

Chiqariluvchi ma`lumotlar:OUTPUT.TXT fayliga qalbaki tangali savat nomerini chop eting.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	10 25 8 1109	2
2	10 25 8 1125	10
3	8000 30 12 959879400	50

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n = 0, w = 0, d = 0, osum = 0, sum = 0; n=sc.nextInt(); //savatlar soni
w=sc.nextInt(); //haqiqiy tanga og`irligi
d=sc.nextInt(); //qalbaki tanga haqiqiy tangadan qancha kamligi
osum=sc.nextInt(); //ajratib olingan tangalar og`irligi
sum = (n*n)/2 + n/2 - n;
//n-1 ta savatdan ajratib olingan tangalar soni, ya`ni s=1+2+..+n-1→ s=n(n-1)/2-n
if (n%2!=0) sum++; sum *= w;
//←barcha ajratilgan tangalar haqiqiy bo`lgandagi og`irlik
//qalbaki tanga joylashgan savatni aniqlash→
if ((sum-osum)/d!=0) pw.print((int) (sum-osum)/d); else pw.print((int) n);
```

№310 Masala

Katakli ramka

To’rtburchak bir xil o’lchamli X×Y ta kvadrat kataklardan tuzilgan. Undan (X-2)×(Y-2) o’lchamli to`rtburchak shunday kesilganki, 1ta katak kengligidagi ramka qoldirilgan. Butun ramkani A×1 o’lchamli plitkalar bilan qoplab bo`ladimi? Plitkalar miqdori cheksiz, ular bir-birining ustiga chiqib ketmaydi va ramka chegarasidan oshib ketmaydi.

**Kiritiluvchi ma`lumotlar: INPUT.TXT** faylining birinchi satrda natural K soni ($1 \le K \le 10$), ya’ni sinovlar soni berilgan . Keyingi K ta qatorda 3 tadan natural son : X, Y – ramka o’lchami, A – plitka o’lchami ($3 \le X, Y \le 2\times10^9$, $1 \le A \le 2\times10^9$) berilgan. Sonlar bo`sh joy bilan ajratilgan.

**Chiqariluvchi ma`lumotlar: OUTPUT.TXT** faylida yagona satrda 0 yoki 1 lardan iborat K ta belgilar chop etilsin(1- mumkin bo`lsasa, 0- aks holda).

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	1 3 3 1	1
2	2 3 3 2 3 3 3	10

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

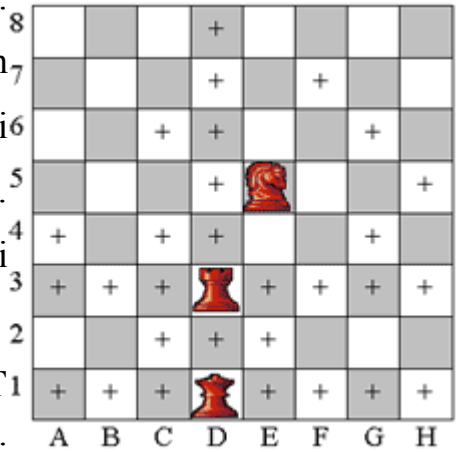
```
int n=sc.nextInt(),t=0,s=0,v,x,y,i,j=0;
//n-sinovlar soni, j-ramka berilgan plitkani joylab bo`lmasligini anglaytadi
int aa[]=new int[n];//a[][]-sinov natijasidan iborat massiv
for(i=0;i<n;i++) { y=sc.nextInt(); x=sc.nextInt(); //x,y –ramka o`lchamlari
v=sc.nextInt(); //v-plitka o`lchami
/*s-ramkaning ajratilgan qismiga joylashtirish mumkin bo`lgan plitkalar soni, t esa
plitka joylangandan keyin qolgan qism*/
s=y/v; t=y%v; //ramkaning 1:y qismiga plitka joylashtirish
if((s!=0)||(t<=1)){ s=(x-1+t)/v; t=(x-1+t)%v; } else j++;
```

```
// ramkaning 1:x-1 qismiga plitka joylashtirish
if((s!=0)||(t<=1)){s=(y-1+t)/v; t=(y-1+t)%v;} else j++;
//← ramkaning 1:y-1qismiga plitka joylashtirish
if((s!=0)||(t<=1)){ s=(x-2+t)/v; t=(x-2+t)%v;} else j++;
// ←ramkaning qolgan 1:x-2 qismiga plitka joylashtirish
if((s!=0)&&(j==0)&&(t==0)) pw.print(1); else pw.print(0); }
```

№19 Masala

Farzin, To`ra va Ot.

8x8 shaxmat doskada 3 ta figura berilgan: farzin, to’ra va ot. Ular yurishi mumkin bo`lgan bo’sh kataklar sonini aniqlang. Figuralar boshqasini ustidan yurishi ham mumkin deb hisoblanadi. Masalan: farzin D5 katakdan turib to’rani urishi mumkin deyiladi.



Kiritiluvchi ma`lumotlar: INPUT.TXT faylida yagona satrda mos ravishda 3 ta figura: farzin, to’ra va otning joylashish koordinatalari berilgan. Koordinatalar lotin harfi(A dan H gacha) va raqam(1 dan 8 gacha) bilan belgilanadi.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida berilgan koordinatalarda turgan figuralar yurishi mumkin bo’lgan kataklar sonini chiqaring.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	D1 D3 E5	29

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
String s1,s2,s3;//s1,s2,s3-farzin,to`ra va otning joylashgan koordinatalari
s1=sc.next();
s2=sc.next();
s3=sc.next();
```

```
char b[]={'A','B','C','D','E','F','G','H'} ;
HashSet a=new HashSet();//ular yurishi mumkin bo`lgan kordinatalar ro`yxati
HashSet d=new HashSet();//ular joylashgan koordinatalar ro`yxati
d.add(s1); d.add(s2); d.add(s3);
int i1=0,i2=0,i3=0,j1=0,j2=0,j3=0,f1,f2,t1,t2,o1,o2,i,j;
i1=s1.charAt(1)-'0'; i2=s2.charAt(1)-'0'; i3=s3.charAt(1)-'0';
//i1,i2,i3-y o`qi bo`yicha, j1,j2,j3 esa x o`qi bo`yicha koordinatalar
char r=s1.charAt(0); for(j=0;j<8;j++) if(b[j]==r) j1=j;
r=s2.charAt(0); for(j=0;j<8;j++) if(b[j]==r) j2=j;
r=s3.charAt(0); for(j=0;j<8;j++) if(b[j]==r) j3=j;
//farzin yurishlarini aniqlash→
f1=i1;f2=j1;
String s="";
while(j1>0){j1--;s=s+b[j1]+i1;a.add(s);s="";}
j1=f2;
while(j1<7){j1++;s=s+b[j1]+i1;a.add(s);s="";}
j1=f2;
while(i1>1){i1--;s=s+b[j1]+i1;a.add(s);s="";}
i1=f1;
while(i1<8){i1++;s=s+b[j1]+i1;a.add(s);s="";}
i1=f1;
while((i1>1)&(j1>0)){i1--;j1--;s=s+b[j1]+i1;a.add(s);s="";}i1=f1;j1=f2;
while((i1<8)&(j1>0)){i1++;j1--;s=s+b[j1]+i1;a.add(s);s="";}i1=f1;j1=f2;
while((i1>1)&(j1<7)){i1--;j1++;s=s+b[j1]+i1;a.add(s);s="";}i1=f1;j1=f2;
while((i1<8)&(j1<7)){i1++;j1++;s=s+b[j1]+i1;a.add(s);s="";}
//to`raning yurishlarini aniqlash→
t1=i2;t2=j2;
while(j2>0){j2--;s=s+b[j2]+i2;a.add(s);s="";} j2=t2+1;
while(j2<8){s=s+b[j2]+i2;j2++;a.add(s);s="";} j2=t2;i2--;
```

```
while(i2>0){s=s+b[j2]+i2;i2--;a.add(s);s="";} i2=t1;
while(i2<8){i2++;s=s+b[j2]+i2;a.add(s);s="";} i2=t1;
//ot yurishlarini aniqlash→
o1=i3;o2=j3;
if((j3>1)&(i3>1)){ s=s+b[j3-2]+(i3-1);a.add(s);s="";}
if((j3>1)&(i3<8)){ s=s+b[j3-2]+(i3+1);a.add(s);s=""; }
if((j3>0)&(i3<7)){ s=s+b[j3-1]+(i3+2);a.add(s);s="";}
if((j3>0)&(i3>2)){ s=s+b[j3-1]+(i3-2);a.add(s);s=""; }
if((i3>1)&(j3<6)) { s=s+b[j3+2]+(i3-1); a.add(s); s=""; }
if((i3<8)&(j3<6)){s=s+b[j3+2]+(i3+1);a.add(s);s=""; }
if((i3>2)&(j3<7)){s=s+b[j3+1]+(i3-2);a.add(s);s=""; }
if((j3<7)&(i3<7)){s=s+b[j3+1]+(i3+2);a.add(s);s="";}
a.removeAll(d);
//ularning o`zlari joylashgan koordinatalarni ro`yxatdan chiqarib tashlash
int k=a.size(); //k ular yurishi mumkin bo`lgan kataklar soni
pw.println(k);
```

№267 Masala

Kserokopiya

Kotiba 1 ta hujjatdan N ta nusxa olishi kerak. Unda 2 ta kseroks bo'lib, biri 1 ta betni x sekundda, 2-si y sekundda nusxalaydi.(1 ta kseroksni, hamda bir vaqtda 2 ta kseroksni ishlatishga ruxsat beriladi. Originaldan ham, nusxasidan ham nusxa olish mumkin). Ishini tugatish uchun eng kamida qancha vaqt ketishini aniqlang.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylida bo'sh joy b-n ajratilgan N, x va y natural sonlari berilgan ($1 \le N \le 2 \cdot 10^8$, $1 \le x, y \le 10$).

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida N ta nusxa olish uchun ketadigan minimal vaqtni(sekundda) aniqlang.

Masalan

№	INPUT.TXT	OUTPUT.TXT

1	4 1 1	3
2	5 1 2	4

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int a=sc.nextInt(),x=sc.nextInt(),y=sc.nextInt(),g=0;
//a-nusxalar soni,x-birinchi kseroks tezligi,y-ikkinchi kseroks tezligi
int m,j; if(x>y){m=x; x=y; y=m;} int s=x,i=0,q=0,t=1,w=0,r=0; a--;
//birinchi marta 1 ta nusxa chiqarib olamiz, uni chiqarishga ketgan vaqt s
q=a*x; //faqat 1-kseroksda q vaqtda nusxalanar edi
while(q>i){ r=i; w=q; i=i+y; q=q-x; }
if(w<i) pw.println(w+s); //2 kseroksni ham ishlatgandagi vaqt
 else pw.println(i+s);//faqat 2- kseroksni ishlatish yetarli
```

№257 Masala

Kub tenglama

$A*X^3 + B*X^2 + C*X + D = 0$ kub tenglamaning barcha butun ildizlarini toping. A,B,C va D –berilgan butun koeffitsientlar.

**Kiritiluvchi ma`lumotlar: INPUT.TXT** faylida A,B ,C va D butun sonlar berilgan. Ular moduli bo`yicha $2*10^9$ dan katta emas.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida avval turli ildizlar sonini, keyin shu ildizlarning o’zini o’sish tartibida joylashtiring. Agar cheksiz ko’p yechim bo’lsa, -1 ni chop eting.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	1 0 0 -27	1 3
2	0 1 2 3	0

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
//g`oya ─kub tenglama yechimlari uning ozod hadi-D ning bo`luvchilaridir
```

```
int i=1,j=0,k=0;//k-tenglamadagi ildizlar soni
boolean t=true,t1=true;
long a=sc.nextInt(),b=sc.nextInt(),c=sc.nextInt(),d=sc.nextInt(),p,x3,x2,z1;
//a,b,c,d- kub tenglamaning koeffitsientlari
long x[]=new long[10000],y1[]=new long[10000],z;
//x[] d ning bo`luvchilaridan , y1[] esa tenglama yechimlaridan iborat massiv
if(d!=0){//tenglamada d nolga teng bo`lmagan hol
//d ning bo`luvchilarini topish→
do{ if(d==i) if(d%i==0) { x[j]=i;j++;x[j]=-1*i;j++;}
if(d!=i) if(d%i==0) { x[j]=i;j++;x[j]=-1*i;j++;
if(i*i!=Math.abs(d)) { x[j]=Math.abs(d/i);j++;x[j]=-1*Math.abs(d/i);j++; } } i++; }
while(i<=Math.sqrt(Math.abs(d)));
for(i=0;i<j;i++){ p=a*x[i]*x[i]*x[i]+b*x[i]*x[i]+c*x[i]+d;
if(p==0) { y1[k]=x[i]; k++; } } }
else { //tenglamada d=0 bo`lgan hol
if((a==0)&(b==0)&(c==0)) { j=-1; t=false; } //cheksiz ko`p yechim
if((a!=0)&(b!=0)&(c==0)) { y1[k]=0;k++; y1[k]=-1*b/a;k++; t1=false; }
//Ax^3+Bx^2=0 ko`rinishidagi tenglama
if(t1){ y1[k]=0; k++; /*Ax^3+Bx^2+Cx=0 tenglamadan x=0 yechimni chiqarib
Ax^2+Bx+C=0 kvadrat tenglamaga o`tish*/
d=c; if(d!=0){ do{ if(d==i) if(d%i==0) { x[j]=i;j++;x[j]=-1*i;j++;}
if(d!=i) if(d%i==0) { x[j]=i;j++;x[j]=-1*i;j++;
if(i*i!=Math.abs(d)) { x[j]=Math.abs(d/i);j++;x[j]=-1*Math.abs(d/i);j++; } } i++; }
while(i<=Math.sqrt(Math.abs(d)));
for(i=0;i<j;i++){ p=a*x[i]*x[i]+b*x[i]+d; if(p==0) { y1[k]=x[i]; k++; } } } } }
if(t) { //topilgan ildizlarni tartiblash
for(i=0;i<k;i++)
for(j=0;j<k-1;j++) { if(y1[j]>y1[j+1]) { z=y1[j];y1[j]=y1[j+1];y1[j+1]=z; } }
pw.print(k+" "); for (i=0;i<k;i++ ) { pw.print(y1[i]+" "); } } if(!t) pw.print("-1");
```

2.2 .Uzun sonlar arifmetikasi (Длинная арифметика)

№40 Masala

2 ning n- darajasi

2^n ning qiymatini aniqlovchi dastur tuzing.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylida yagona satrda n natural son berilgan (0 < n < 1000).

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida 2^n ning qiymati chiqarilsin.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	3	8
2	10	1024
3	72	4722366482869645213696

Qo`yilgan masalaning dasturi Java dasturlash tilida quyidagicha bo`ladi:

```
import java.util.*; import java.math.*; import java.io.*;
class Main{ public static void main(String mm[]) throws IOException{
Scanner sc=new Scanner(new File("input.txt"));
PrintWriter pw=new PrintWriter(new File("output.txt"));
int n=sc.nextInt(),i; /*bu yerda: n- 2 ning hisoblanishi kerak bo`lgan darajasi;
i- 1 dan n gacha 1 qadam bilan darajani hisoblash */
BigInteger s=BigInteger.valueOf(1);
/* ikkining n-darajasi, uni BigInteger sinfining elementi sifatida e`lon qilib, uning
qiymati birga teng*/
for(i=1;i<=n;i++) s=s.multiply(BigInteger.valueOf(2));
//s ga ikkini ko`paytirib boriladi, n marta.
pw.println(s); // s ni chop eting
pw.close(); } }
```

Keyingi masalalarning dasturini faqat asosiy qismini keltiramiz, ya`ni barcha masalalarda dastur tuzilishi quyidagicha bo`ladi. Faqat asosiy qism o`zgaradi.

```
import java.util.*; import java.io.*; import java.math.*;
class Main{ public static void main(String mm[]) throws IOException {
Scanner sc=new Scanner(new File("input.txt"));
PrintWriter pw=new PrintWriter(new File("output.txt"));
```

Asosiy qism

```
pw.close(); } }
```

№286 Masala

Kattami yoki Kichik–2

Kiritiluvchi ma`lumotlar: **INPUT.TXT** fayli ikkita satrdan iborat bo'lib, ularning har birida 1 tadan haqiqiy son oldidagi nollarsiz berilgan. Sonlarning butun va kasr qismi nuqta bilan ajratilgan. Agar butun son bo'lsa nuqta ishlatilmaydi. Har bir sonning raqamlari (ya'ni xonalari soni) 10000 tadan ko'p emas.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida 1 ta belgini chop eting:agar 1-son 2-sidan kichik bo'lsa,"<" belgisini; katta bo'lsa, ">" belgisini, 2lasi teng bo'lsa, "=" belgisi chiqarilsin.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	2.39 3.61	<
2	123 12.3	>
3	12345678 12345678.0	=

4	-1.0 1.0	<

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

String x,y; x=sc.next(); y=sc.next(); BigDecimal a=new BigDecimal(x); BigDecimal b=new BigDecimal(y); if(a.compareTo(b)==1) pw.println(">"); if(a.compareTo(b)==0) pw.println("="); if(a.compareTo(b)==-1) pw.println("<");

№109 Masala

Bo`linma(A / B)

2 ta natural sonlar A va B uchun A/B bo’linmaning aniq qiymatini topish talab etiladi.

**Kiritiluvchi ma`lumotlar: INPUT.TXT** faylida 1000 dan oshmaydigan 2 ta son orasida “/”(bo’lish) belgisi bilan kiritilgan. Ortiqcha bo`sh joy ishlatilmagan.

**Chiqariluvchi ma`lumotlar: OUTPUT.TXT** faylida A/B bo`linmaning aniq qiymatini ortiqcha nuqtalar, nollar yoki bo`sh joylarsiz chop etish lozim. Cheksiz o’nli yozuvda davrni qavslar ichiga yozib chiqaring.

Masalan: 08.92, 3.20, 120.6(6), 0.(33), 5.(0), 2. , .3, 0.33(03) sonlari noto’g’ri chop etilgan deb hisoblanadi. Ularni to’g’ri ko’rinishda 8.92, 3.2, 120.(6), 0.(3), 5, 2, 0.3, 0.3(30) kabi chop etish lozim.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	10/7	1.(428571)
2	1/3	0.(3)
3	100/25	4

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

String S="",x1=sc.next();//x1-berilgan satr,S hosil qilinishi kerak bo`lgan bo`linma boolean ry=true;double bolinma;

```
int l=x1.length(),i=0,k=0,a,a1=0,p,q,x,y,z=0,b,b1,j=0,c=0;
BigInteger s=new BigInteger("10");
for(i=0;i<l;i++) { if(x1.charAt(i)=='/') {k=i;break;} }
//k-"/" ning indeksi, a,b bo`linmadagi A va B sonlar
a=Integer.parseInt(x1.substring(0,k));b=Integer.parseInt(x1.substring(k+1,l));
if(a<b) c=a; else c=b; j=2;
while(j<=c) { while((b%j==0)&(a%j==0)) {a=a/j;b=b/j;} j++;}
//kasrni qisqartirish
if(b==1) { pw.println(a); ry=false; }//qisqartirish natijasida maxraj bir bo`lsa
j=1; x=0; y=0; p=1; b1=b;
while (j<=b1) { j++; while(b1%j==0) {
if(j==2) x++; if(j==5) y++; if ((j!=2)&(j!=5)) p=p*j; b1=b1/j; } }
//←maxraj bir bo`lmasa uning bo`luvchilarini topish, x,y — 2 va 5 lar soni
if((p==1)&(ry)) { bolinma=(double)a/b; pw.println(bolinma); ry=false; }
// o`nli kasr bo`lgan hol
if(x<y) x=y; q=1; /* x va y dan kattasini olamiz, aralash davriy kasrda qavsdan
oldin shuncha raqam bo`ladi*/
while(s.subtract(BigInteger.valueOf(1)).mod(BigInteger.valueOf(p)).compareTo
(BigInteger.valueOf(0))!=0) { q++; s=s.multiply(BigInteger.valueOf(10)); }
//←kasrning davridagi raqamlar sonini toppish,q-davrdagi raqamlar soni
if(a>b) { a1=a/b; a=a-a1*b; } S=S+a1+"."; //a1-A/B ning butun qismi
i=1; if (x!=0) while (i<=x) { a=a*10; z=a/b; S=S+z; i++; a=a-b*z; }
//←kasr aralash bo`lsa davrdan oldingi "." dan keyingi raqamlarni topish
S=S+"("; i=1; while (i<=q) { a=a*10; z=a/b; S=S+z; a=a-b*z; i++; } S=S+")";
//←davrdagi raqamlarni topish
if (ry) pw.print(S);
```

№367 Masala

Daraja-2

2 ta natural A va B sonlari berilgan . A^B ni hisoblovchi dastur tuzing.

**Kiritiluvchi ma`lumotlar: INPUT.TXT** faylida yagona satrda bo`sh joy bilan ajratilgan A va B sonlari berilgan, bunda ($1 <= A <= 9$, $1 <= B <= 10^4$).

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida darajaga ko'tarish natijasini oldida keluvchi nollarsiz chop eting.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	3 20	3486784401
2	2 16	65536
3	5 50	88817841970012523233890533447265625

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int a,b,i; a=sc.nextInt(); b=sc.nextInt(); BigInteger s=BigInteger.valueOf(1);
for(i=1;i<=b;i++) { s=s.multiply(BigInteger.valueOf(a));} pw.print(s);
```

№311 Masala

Faktoriallar yig'indisi.

K sonning faktoriali K!=1×2×3×…×K.ga teng.

Berilgan N soni bo'yicha 1!+2!+…+N! faktoriallar yig'indisini topuvchi dastur tuzing.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylida N (N<=200) natural soni berilgan.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida natija chiqarilib, u hisoblanayotgan yig'indining barcha o'nlik belgilariga ega bo'lsin.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	1	1
2	2	3
3	3	9

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n=sc.nextInt(),i,j; BigInteger s=BigInteger.valueOf(0);
BigInteger p=BigInteger.valueOf(1);
for(i=1;i<=n;i++) { for(j=1;j<=i;j++) p=p.multiply(BigInteger.valueOf(j));
s=s.add(p); p=BigInteger.valueOf(1); } pw.print(s);
```

№363 Masala

Uzun ko`paytma

2 ta nomanfiy butun M va N sonlari berilgan. Ularning ko'paytmasini topuvchi dastur tuzing.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylida 1-satrda M, 2-satrda N sonlari yozilgan. (0 <= M, N <= 102500)

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida M*N ko'paytmani chiqaring.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	5 7	35
2	4134937827592 784	3241791256832128
3	9876543210 1023456789	10108215200126352690

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
String x,y;x=sc.next();y=sc.next(); BigInteger b; BigInteger n=new BigInteger(x);
BigInteger k=new BigInteger(y); b=n.divide(k); pw.print(b);
```

№452 Masala

Fibonachchi sanoq tizimi

Fibonachchi sonlari quyidagi boshlang'ich qiymatlar va munosabat bo'yicha aniqlanadi: U1=1; U2=2; Un=Un-1+Un-2.

Shunday 0 va 1 lar qatnashuvchi sanoq tizimini ko'rib chiqamizki , 2 lik sanoq tizimidan farqli ravishda ularda asosiy deb 2 ning darajalari bo'lgan 1,2,4,8,16,… emas, balki Fibonachchi sonlari 1,2,3,5,8,13,… qaraladi.

Bu tizimda har bir musbat butun son yagona usul bilan nol va birlardan tuzilgan shunday satr sifatida qaraladiki, bu satr 1 raqami bilan boshlanadi va 2 ta 1 raqami hech qachon yonma-yon kelmaydi.

A va B sonlarining Fibonachchi sanoq tizimida ifodalangan ko`rinishi 2 ta satrda berilgan. A+B sonini shu tizimda ifodalovchi satrni topuvchi dastur tuzing.

Masalan, ushbu boshlang'ich satrlar 10101 va 100 mos ravishda 1*8+0*5+1*3+0*2+1*1=8+3+1=12 va 1*3+0*2+0*1=3 sonlarini ifodalaydi. Javob sifatida 100010 satr olinadi, u 1*13+0*8+0*5+0*3+1*2+0*1=13+2=15=12+3 sonini ifodalaydi.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylida 1-qatorda A soni va 2-qatorda B soni yozilgan. A, B sonlari hamda A+B yig'indining Fibonachchi sanoq tizimidagi yozuvi uzunligi 255 ta belgidan oshmaydi.

**Chiqariluvchi ma`lumotlar: OUTPUT.TXT** faylida A+B yig`indining Fibonachchi sanoq tizimidagi ko'rinishini chiqaring. Yechimlar bir nechta bo'lsa, ixtiyoriysini chop etish mumkin.

Masalan

№	**INPUT.TXT**	**OUTPUT.TXT**
1	10101 100	100010

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

String a,b,q=""; a=sc.next(); b=sc.next(); /*a,b fibonachchi sanoq tizimidagi sonalar,q-ular yig`indisining fibonachchi sanoq tizimidagi ko`rinishi */

```
BigInteger F[]=new BigInteger [255];//F[]-fibonachchi sonlaridan iborat massiv
int x,y,i,j,k; BigInteger z,h,p=BigInteger.valueOf(0),s=BigInteger.valueOf(0);
F[0]=BigInteger.valueOf(1);F[1]=BigInteger.valueOf(2);
for (i=2;i<255;i++) F[i]=F[i-1].add(F[i-2]);//255 ta fibonachchi sonlarni topib olish
x=a.length();y=b.length();i=0;
for (j=x-1;j>=0;j--) { z=BigInteger.valueOf(a.charAt(j)-'0');
h=z.multiply(F[i]); s=s.add(h); i=i+1; } i=0;
//birinchi sonni o`nli sanoq tizimiga o`tkazish
for (j=y-1;j>=0;j--) { z=BigInteger.valueOf(b.charAt(j)-'0');
h=z.multiply(F[i]); p=p.add(h); i=i+1; }
//ikkinchi sonni o`nli sanoq tizimiga o`tkazish, s shu sonlar yig`indisi
s=s.add(p); i=0; k=0;
while ((F[k].compareTo(s)==-1)|(F[k].compareTo(s)==0)) k=k+1;
//k- s dan kichik yoki teng fibonachchi sonining indeksi
for (i=k;i>=0;i--) { if ((F[i].compareTo(s)==0)|(F[i].compareTo(s)==-1))
{ s=s.subtract(F[i]); q=q+1; } else if (i<k) q=q+0; } pw.print(q);
```

№474 Masala

Keane ketma-ketligi

Keane tomonidan taqdim etilgan cheksiz ketma-ketlik 0010011100010011101101100 01… ga teng bo’lib,quyidagi algoritm yordamida topiladi: dastlab 0, keyin 001 yoziladi, so’ngra 001001110, ya’ni ketma-ketlikning keyingi hadini topish uchun oldingi hadi 2 marta yoziladi va o’ng tomondan oldingi hadning teskarisi(manfiyi) yozib boriladi. Ushbu qatorning elementlari boshlang’ich Keane qism ketma-ketliklari hisoblanadi.

Berilgan N soni bo’yicha shu ketma-ketlikning N-nomerdagi Bit qiymatini topuvchi dastur tuzing.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylida N ($N<=10^{200}$) soni berilgan.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida qidirilayotgan bit qiymati chiqarilsin.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	3	1
2	18	0

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
BigInteger t[]=new BigInteger[1000];
int i=0,j=2,c=-3,y=2; t[0]=new BigInteger("3");
String s=sc.next();//s- berilgan sonning satr ko`rinishi
BigInteger a=new BigInteger(s),b=new BigInteger("3"),r=new BigInteger("1"),w=new BigInteger("0"),k=new BigInteger("-1"),l=new BigInteger("0"),z;//a- s ning sonli ko`rinishi
while(c<1) { c=b.pow(j).compareTo(a); if(c>0) break;
 i++; t[i]=b.pow(j); j++; if(c==0)y=0;}
//a dan katta bo`lgan 3 ning darajasini topish,agar a=3^j bo`lsa y=0 bo`ladi
j=0; c=1; z=new BigInteger("2"); l=new BigInteger("0"); int x=1; c=2;
//a<3^j bo`lgan hol
if(y!=0) { a=new BigInteger(s);
while(i!=-1) { c=a.compareTo(t[i].multiply(BigInteger.valueOf(2)));
// a>2*3^i bo`lgan hol→
if(c>0){ a=a.subtract(t[i].multiply(BigInteger.valueOf(2))); j++; i--; }
// a>2*3^i bo`lgan hol, a>// a>3^i bo`lsa a=a-3^i
else { c=a.compareTo(t[i]); if(c>0) { a=a.subtract(t[i]); i--; } else i--; }
c=a.compareTo(BigInteger.valueOf(3)); if(c<=0)break;
//jarayon a<=3 bo`lguncha davom etadi
 } }
String m=a.toString(); x=Integer.parseInt(m);
if(x==3)x=0; //a 3 ning biror darajasi bo`lgan hol
```

```
if(y==0) { if(i%2==0) pw.println("1"); else pw.println("0");}
else //a dan 3 ni darajalarini ayirish natijasida u 3 ga teng bo`lsa
if(x==0){ if(j%2==0) pw.println("1"); else pw.println("0"); }
// a dan 3 ni darajalarini ayirish natijasida u 3 dan kichik bo`lgan hol
else { if(j%2==0) pw.println("0"); else pw.println("1"); }
```

№75 Masala

Ko'paytmalarning yi'g'indisi

Barcha N xonali sonlarning raqamlari ko'paytmasining umumiy yig'indisini topish talab etiladi. Agar biror sonda 0 raqami uchrasa, u holda uning raqamlari ko'paytmasi ham 0 ga teng deb hisoblanadi. N=3 uchun izlanayotgan yig'indi quyidagicha aniqlanadi: (100 dan 999 gacha bo'lgan sonlar uchun)

S = 1*0*0 + 1*0*1 + 1*0*2 + … + 9*9*8 + 9*9*9 = 91125

Kiritiluvchi ma`lumotlar: INPUT.TXT faylida N (N<1000) natural soni berilgan.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida shu N xonali sonlar raqamlari ko'paytmalarining umumiy yig'indisi chop etilsin.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	1	45
2	3	91125
3	5	184528125

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int x=sc.nextInt(); BigInteger s; s=BigInteger.valueOf(45).pow(x);
/*1 dan 9 gacha bo`lgan raqamlar yig`indisi, ya`ni bir xonali sonlar yig`indisi 45 ga teng, 2 xonali sonlarning raqamlari ko`paytmalari yig`indisi esa 45*45 ga,... n xonali sonlarning raqamlari ko`paytmalari yig`indisi 45 ning n- darajasiga teng */
 pw.println(s);
```

№146 Masala

Uzun ildiz

Berilgan natural A soni bo'yicha B^2 <= A shartga javob beruvchi eng katta B sonini topish talab etiladi.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylida A natural soni (A <= 103000) berilgan.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida kvadrati A dan oshmaydigan eng katta B natural sonini chop eting. B soni oldinda keluvchi nollarsiz chiqarilsin.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	17	4

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
String xx; xx=sc.next(); int d=1,l; l=xx.length(); l=l/2+1;
BigInteger x=new BigInteger(xx.substring(0,l)),y;
//x xx ning 0 dan l gacha bo`lgan qismi
BigInteger a=new BigInteger(xx); //a berilgan son
while(d==1){ y= a.divide(x); y= x.add(y); y=y.divide(BigInteger.valueOf(2));
d=(y.pow(2)).compareTo(a);
//y^2 ning a ga tengligini tekshirish, jarayon y aning ildizi bo`lguncha davom etadi
x=y; } pw.println(x);
```

2.3. Ikki o`lchovli massivlar(Двумерные массивы)

№ 446 Masala

Tablo.

Bir shaharning xokkey stadionida katta to'g'ri to'rtburchak shaklidagi tablo (hisob yoziladigan doska) bor. U n ta satr va m ta ustunga (ya'ni m*n ta

yacheykalarga) ega. Xokkey matchi davomida bu tabloda hisob va vaqt ko'rsatilsa, tanaffuslarda reklama beriladi.

Shu sababli tabloda reklama berib bo'ladimi yoki yo'qmi – ushbu imkoniyatni tekshirish masalasi vujudga keldi. Reklama ham n ta satr m ta ustunda tasvirlanadi. Har bir katakchasi 4 xil rangdan biriga bo'yalgan. 3 ta asosiy qizil – R, yashil – G, ko'k – B hamda qora - " ."(nuqta).

Tabloning har bir yacheykasi o'zining rang uzatish imkoniyatlari bilan farqlanadi. Ixtiyoriy yacheyka qora rangni ko'rsata oladi. Shuningdek, yacheykalarning har biri asosiy ranglardan bir qismini ko'rsata oladi.

Bu quyidagicha kod bilan ifodalanadi:

0 – yacheyka faqat qora rangni ko'rsata oladi.

1 – faqat qora va ko'k rangni ko'rsata oladi.

2 – qora va yashil rangni ko'rsata oladi.

3 – qora, yashil va ko'k rangni ko'rsata oladi.

4 – qora va qizil rangni ko'rsata oladi.

5 – qora, qizil va ko'k rangni ko'rsata oladi.

6 – qora, qizil va yashil rangni ko'rsata oladi.

7 – qora, qizil, yashil, ko'k rangni ko'rsata oladi.

Tabloning tavsifi va reklama bayoniga ko'ra , tabloda shu reklamani tasvirlab bo'lish bo'lmasligini aniqlang.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylining 1- qatorda n va m sonlari berilgan. Keyingi n ta qatorda m tadan belgi –reklama tavsifi keltirilgan bo'lib, belgilarning har biri { R, G ,B } ranglar to'plamiga tegishli. Shundan so'ng tablo tavsifi keltirilgan bo'lib, u ham n ta qatorda m tadan son orqali beriladi. Sonlar kodlari yuqorida keltirildi.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida natijani "YES" yoki "NO" javoblaridan biri orqali chop eting.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	3 3 .GB R.B RG. 0 1 2 3 4 5 6 7 0	NO
2	2 3 RGB .G. 7 7 7 7 7 7	YES

Qo`yilgan masalaning dasturi Java dasturlash tilida quyidagicha bo`ladi:

```
import java.util.*; import java.io.*;
class Main { public static void main (String arg[]) throws IOException {
Scanner sc=new Scanner(new File("input.txt"));
PrintWriter pw=new PrintWriter(new File ("output.txt"));
String s; int n,m,w=0,i,j,v; n=sc.nextInt(); m=sc.nextInt();
char a[][]=new char[n][m]; //a[][]-reklama tavsifidan iborat massiv
for(i=0;i<n;i++) { s=sc.next(); for(j=0;j<m;j++)a[i][j]=s.charAt(j); }
for(i=0;i<n;i++) for(j=0;j<m;j++){ v=sc.nextInt(); //tablo yacheykasi tavsifi
//keying qatorlarda reklama tavsifi va yacheyka tavsifining mos kelishini tekshirish
if(v==0) { if((a[i][j]=='.')) w++; } else
if(v==1) { if((a[i][j]=='.')||(a[i][j]=='B')) w++; } else
if(v==2) { if((a[i][j]=='.')||(a[i][j]=='G')) w++; } else
if(v==3) { if((a[i][j]=='.')||(a[i][j]=='B')||(a[i][j]=='G' ))w++; } else
if(v==4) { if((a[i][j]=='.')||(a[i][j]=='R'))w++; } else
if(v==5) { if((a[i][j]=='.')||(a[i][j]=='R')||(a[i][j]=='B' ))w++; } else
```

```
if(v==6) { if((a[i][j]=='.')||(a[i][j]=='R')||(a[i][j]=='G' ))w++; } else
if(v==7) { if((a[i][j]=='.')||(a[i][j]=='R')||(a[i][j]=='B')||((a[i][j]=='G'))) w++; } }
if(w==n*m)pw.print("YES"); else pw.print("NO");
pw.close(); } }
```

Keyingi masalalarning dasturini faqat asosiy qismini keltiramiz, ya`ni barcha masalalarda dastur tuzilishi quyidagicha bo`ladi. Faqat asosiy qism o`zgaradi.

```
import java.util.*; import java.io.*; import java.math.*;
class Main{ public static void main(String mm[]) throws IOException {
Scanner sc=new Scanner(new File("input.txt"));
PrintWriter pw=new PrintWriter(new File("output.txt"));
```

Asosiy qism

```
pw.close(); } }
```

№ 493 Masala

Dengiz o'yini – 3.

"Dengiz o'yini " – 2 ta ishtirokchili o'yin bo'lib,o'yinchilar navbati bilan raqiblarning o'zlariga noma'lum bo'lgan xaritalaridan koordinatalarni aytadilar. Agar bu koordinatalarda raqibning kemalari joylashgan bo'lsa, kema yoki qismi "cho'kadi", cho'ktirgan o'yinchi esa yana 1 ta yurish qilish huquqiga ega bo'ladi. O'yinchilarning maqsadi – birinchi bo'lib raqiblarning kemalarini cho'ktirishdir.

Innokentiy va Yemelyan yangi o'yin boshladilar. Ularning o'yin qoidalari an'anaviy qoidalardan biroz farq qiladi.

1-dan, o'yin 10*10 maydonda emas, N*M o'lchamli maydonda o'tkaziladi.

2-dan, kemalar soni , o'lchami va shakli bolalar tomonidan o'yin oldidan aiqlanadi.

– shunda o'yin qiziqroq kechadi.

Yemelyan o'zining 1 tadan tashqari barcha kemalarini joylashtirib chiqdi. Qolgan 1 ta palubali kemani necha xil usulda joylashtirish mumkinligini aniqlang. Bunda hisobga olish kerakki,qoidalarga ko'ra kemani shunday katakka

joylashtirish lozimki, unga qo'shni barcha kataklar band bo'lmasligi kerak. Qo'shni kataklar deganda 1 ta umumiy tomonga ega kataklar tushiniladi.

1 ta palubali kema 1 ta katakka joylashadi.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylining 1- qatorda N va M sonlari berilgan. Keyingi N ta qatorda o'yin maydoni bayon qilingan bo'lib, har bir qatotda M tadan belgi beriladi.

"." (nuqta) –bo'sh katakni, "*" –kema bilan band bo'lgan katakni bildiradi.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida variantlar sonini chop eting.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	4 4 **** **.. *... *...	4
2	4 3 *** ... ... ***	0

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
String s=""; int n=sc.nextInt(),j,m,i,w=0,r=0;
/*w-yulduzchalar soni,m,n- o`yin maydonining o`lchami, r-palubali kemani joylashtirish mumkin bo`lgan koordinatalar soni */
m=sc.nextInt();
char a[][]=new char[n][m];//a[][]-o`yin maydonidan ibrat massiv
for(i=0;i<n;i++) { s=sc.next(); for(j=0;j<m;j++) a[i][j]=s.charAt(j); }
```

```
for(i=0;i<n;i++){ for(j=0;j<m;j++){ w=0; if(a[i][j]=='.') {
if(i+1<n){ if(a[i+1][j]=='*') w++; }
//qaralayoygan koordinatadan pastdagi koordinatani tekshirish
if(j+1<m){ if(a[i][j+1]=='*') w++; }
//qaralayotgan koordinatadan o`ngdagi koordinatani tekshirish
if(i-1>=0){ if(a[i-1][j]=='*') w++; }
// qaralayotgan koordinatadan yuqoridagi koordinatani tekshirish
if(j-1>=0){ if(a[i][j-1]=='*') w++; }
// qaralayotgan koordinatadan chapdagi koordinatani tekshirish
if(w==0)r++;//demak ushbu koordinataga qo`shni koordinatada yulduzcha yo`q
} } }
pw.print(r);
```

№58 Masala

Xushro'ylikni tekshirish

n ta satr va m ta ustunga ega, har bir katagida 0 yoki 1 yozilgan jadval berilgan. Jadval xushro'y deyilishi uchun unda 2*2 o'lchamdagi faqat 0 lar yoki faqat 1 lardan iborat jadvallar bo'lmasligi kerak.

Masalan: chapdagi 4*4 jadval xushro'y, o'ngdagi 3*3 o'lchamli jadval – xushro'y emas.

1	0	1	0
1	1	1	0
0	1	0	1
0	0	0	0

0	0	**1**
0	0	**1**
1	1	**1**

Bir necha jadval berilgan. Ularning xushro'y yoki xushro'y emasligini aniqlang.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylining 1-qatorida t-jadvallar soni berilgan. Jadvallar quyidagicha tavsiflanadi: har bir jadval uchun dastlab 2 ta - n va m sonidan iborat qator, so'ngra n ta qator keltirilib, ularning har birida m tadan son beriladi. Shu jadval tavsifidagi (i+1)-satrning j-soni – mos jadvalning a_{ij} elementidir. Barcha a_{ij} lar 0 yoki 1 ga teng.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida har bir jadvalning xushro'y yoki xushro'y emasligiga qarab mos ravishda "YES" yoki "NO" javoblarini chiqaring.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	3 1 1 0 4 4 1 0 1 0 1 1 1 0 0 1 0 1 0 0 0 0 3 3 0 0 1 0 0 1 1 1 1	YES YES NO

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n,i,j,l,nn,m,f=0;
n=sc.nextInt ();//n-jadvallar soni,f-aytilgan shakldagi kvadratlar soni
for(i=0;i<n;i++) { //nn,m- i- jadval o`lchamlari
nn=sc.nextInt (); m=sc.nextInt ();
int aa[][]=new int[nn][m]; //a[][] i- jadval ,pastda uni o`qib olish
for(j=0;j<nn;j++) { for(l=0;l<m;l++) aa[j][l]=sc.nextInt(); }
for(j=0;j<nn;j++){ for(l=0;l<m;l++)
if((l+1<m)&&(j+1<nn)&&(aa[j][l]==0)&&(aa[j+1][l]==0)&&(aa[j][l+1]==0)
&&(aa[j+1][l+1]==0)) f++; //nollardan iborat massiv mavudligini tekshirish
if((l+1<m)&&(j+1<nn)&&(aa[j][l]==1)&&(aa[j+1][l]==1)&&(aa[j][l+1]==1)
```

```
&&(aa[j+1][l+1]==1)) f++; //birlardan iborat massiv mavudligini tekshirish
}
if(f==0) pw.println("YES"); else pw.println("NO");
f=0; }
```

№ 27 Masala

Rassom.

Eni w va bo'yi h bo'lgan oq rangdagi to'g'ri to'rtburchak shaklidagi mato olib, unda bir necha n ta to'g'ri to'rtburchak chizildi. Bu to'g'ri to'rtburchaklarning tomonlari katta to'rtburchak tomonlariga parallel va uchlari butun sonli koordinatalarda joylashgan. Matoning chizilmagan(bo'yalmagan) qismi yuzasini toping.

**Kiritiluvchi ma`lumotlar: INPUT.TXT** faylining 1-qatorda w va h- mato o`lchamlari berilgan. 2-qatorda n– to'rtburchaklar soni, keyingi n ta qatorda kichik to'rtburchaklar haqida ma'lumotlar –x1,y1,x2,y2, ya'ni chap yuqori va o'ng pastki uchlari koordinatalari keltirilgan.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida bo'yalmagan yuzani chop eting.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	5 5 2 1 1 3 3 2 2 4 4	18
2	6 7 3 0 0 5 5 1 1 4 4 2 2 3 3	17

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n,i,j,jj; HashSet b1=new HashSet (); int x1,y1,x2,y2,x3,y3,ss=0,s=0,l;
String scc="";
//x1,y1-to`rtburchak mato koordinatalari,n-bo`yalgan to`rtburchaklar soni
x1=sc.nextInt(); y1=sc.nextInt(); n=sc.nextInt(); int aa[][]=new int[x1][y1];
for(i=0;i<n;i++) { //x2,y2,x3,y3-bo`yalgan to`rtburchaklar koordinatalari
x2=sc.nextInt(); y2=sc.nextInt(); x3=sc.nextInt(); y3=sc.nextInt();
for(j=x2;j<x3;j++) { for(l=y2;l<y3;l++) aa[j][l]=1; } }
//bo`yalgan qismlarni birlar bilan to`ldirish
for(j=0;j<x1;j++) { for(l=0;l<y1;l++) if(aa[j][l]==0) ss=ss+1; }
//ss-bo`yalmagan koordinatalar soni
pw.print(ss);
```

№ 265 Masala.

Shaxmat taxtasi.

Shaxmat taxtasidan kataklar chegarasi bo'yicha qismlarga ajralmaydigan, kovaklarga ega bo'lmagan shakl qirqib olindi. Uning perimetrini toping.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylining 1-qatorda N-qirqib olingan kataklar soni, keyingi n ta qatorda kataklar koordinatalari(1 dan 8 gacha – satr ham ,ustun ham shunday nomerlanadi) berilgan. Har bir qirqiluvchi katak 1 marta ko'rsatiladi.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida qirqib olingan shakl perimetrini(katak tomoni 1 ga teng) chop eting.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	3	8

	1 1 1 2 2 1	
2	1 8 8	4

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n,i,j,m=0,m1=0;
//n-qirqib olingan kataklar soni,m,m1-qirqib olingan katak koordinatalari
int b[][]=new int[9][9]; int a[][]=new int[9][9]; /*8 ta katak 9 ta chiziqni kesishdan hosil qilinganligi uchun massivlar [9][9] o`lcvhamda olinadi(har bir katakda 2 ta bo`yi va 2 ta eni borligi ughun massiv 2 o`lchovli)*/
n=sc.nextInt();
for(i=0;i<n;i++){ m=sc.nextInt(); m1=sc.nextInt();
/*a[][] va b[][] massivlar qirqilgan katakning bo`yi va enidagi chiziqlarni ifodalaydi. Agar katak qirqilgan bo`lsa qirqilgan shakl perimetrini hisoblash uchun shu katakning vaziyati(bu katak boshqa qirqilgan katakga yopishganmi yoki 1 ta katak qirqib olinganmi) hisobga olinadi*/
if(a[m-1][m1-1]==1) a[m-1][m1-1]=0; else a[m-1][m1-1]=1;
if(a[m-1][m1]==1) a[m-1][m1]=0; else a[m-1][m1]=1;
if(b[m1-1][m-1]==1) b[m1-1][m-1]=0; else b[m1-1][m-1]=1;
if(b[m1-1][m]==1) b[m1-1][m]=0; else b[m1-1][m]=1;
//Natija:qirqilgan shakl chegarasini ifodalovchi massivlar elementlari 1 ga teng
}
int s=0; //s- qirqib olingan katak perimetri
for(i=0;i<9;i++) for(j=0;j<9;j++) { s=s+a[i][j]+b[i][j]; }
pw.println(s)
```

№ 196 Masala

Spiral.

Kvadrat matritsani rasmda ko'rsatilganidek spiral shaklida to'ldirib chiqish kerak: bu ish chap yuqori burchakda 1 ni yozishdan boshlanib, markazda N^2 sonini yozish bilan tugaydi, N- matritsa tartibi.

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

**Kiritiluvchi ma`lumotlar: INPUT.TXT** faylida natural son N(N<=100)-matritsa o`lchami berilgan .

**Chiqariluvchi ma`lumotlar: OUTPUT.TXT** faylida spiral shaklida to`ldirilgan N*N matritsani chop eting, bunda sonlar orasida ixtiyoriy sondagi bo'sh joy bo'lishi mumkin.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	5	1 2 3 4 5 16 17 18 19 6 15 24 25 20 7 14 23 22 21 8 13 12 11 10 9

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n=sc.nextInt();//kvadrat matritsa o`lchami
int j=0,i=0,v=0,x,y,x1,t=0,h;
int a[][]=new int[n][n];//a[][] hosil qilinadigan spiral shaklidagi matritsa
h=0; x=n; //h─1 dan N^2 gacha bo`lgan sonlar
//a[][] massivni 1 dan N^2 gacha bo`lgan sonlar bilan to`ldirish
while(h<n*n) { j=t; i=t;
while(j<x) { h++; a[i][j]=h; j++; } //chapdan o`ngga satr bo`ylab yurish
j--; //shart bajarilmagan holatda j 1 taga oshadi, shu sababli 1 taga kamaytiramiz
i++; //1 satr pastga tushish
while(i<x) { h++; a[i][j]=h; i++; } //yuqoridan pastga ustun bo`ylab yurish
i--; //shart bajarilmagan holatda i 1 taga oshadi, shu sababli 1 taga kamaytiramiz
```

```
j--; //chapga 1 ustun siljish
while(j>=t) { h++; a[i][j]=h; j--; } //o`ngdan chapga qarab yurish
j++; //shart bajarilmagan holatda j 1 taga kamayadi, shu sababli 1 taga oshiramiz
i--; //yuqoriga 1 satr siljish
while(i>t) { h++; a[i][j]=h; i--; } //pastdan yuqoriga qarab yurish
t++;
//yuqori o`ng burchakdan 1 ustun va 1 satr to`ldi, shuning uchun t 1 taga ortadi
x--;
 //quyi chap burchakdan 1 ustun va 1 satr to`ldi, shuning uchun x 1 taga kamayadi
 }
for(i=0;i<n;i++) { pw.println(); for(j=0;j<n;j++) pw.print(a[i][j]+" "); }
```

№ 197 Masala

Iloncha.

N*N matritsani rasmdagidek iloncha shaklida to'ldiring, bu ish chap yuqori burchakda 1 ni yozishdan boshlanib, pastki o'ng burchakda N^2 sonini yozish bilan tugaydi, N- matritsa tartibi.

1	3	4	10
2	5	9	11
6	8	12	15
7	13	14	16

**Kiritiluvchi ma`lumotlar: : INPUT.TXT** faylida natural son N(N<=100)- matritsa o`lchami berilgan .

**Chiqariluvchi ma`lumotlar: OUTPUT.TXT** faylida iloncha shaklida to`ldirilgan N*N matritsani chop eting, bunda sonlar orasida ixtiyoriy sondagi bo'sh joy bo'lishi mumkin.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	4	1 3 4 10 2 5 9 11 6 8 12 15 7 13 14 16

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n=sc.nextInt(); //n- kvadrat matritsa o`lchami
int a[][]=new int[n][n]; //a[][]-iloncha shaklidagi matritsa
int i=0,k=0,k1=0,t1=0,j=0,h=0,t=0,l,p,p1;
/*h-1 dan N2 gacha bo`lgan sonlar,k1-to`ldirilayotgan yo`lak raqami(toq bo`lsa pastdan yuqoriga, aks holda yuqoridan pastga yo`nalish), i-yo`lak boshlanadigan satr indeksi,k-yo`lak boshlanadigan ustaun indeksi, t, j-to`ldirilayotgan element indekslari, p, p1 to`ldirilayotgan element qiymatini aniqlash uchun foydalaniladi*/
while(h<n*n) { t=i;
while(j<=i) { if(k1%2==1) { p=0;p1=0; } else { p=t;p1=j; }
h++; a[t][j]=h-p1+p; t--; j++; }
if(i<n-1) { i++;j=0; } else { k++; j=k; } k1++; }
for(i=0;i<n;i++) { for(j=0;j<n;j++) pw.print(a[i][j]+" "); pw.println(); }
```

№ 290 Masala

Terrorchilar bazasi.

Maxsus xizmat xodimlari cho’lning borish qiyin qismida terrorchilarning yaxshi niqoblangan bazasi mavjudligi haqida xabar oldilar. Ular terrorchilar bazasining plani(rejasi)ni ham qo’lga kiritdilar. Ushbu baza yuqoridan cho’lda ko’p uchraydigan qoyatosh bo’laklari ko’rinishiga ega. Bu bazani uchar raketalar bilan yo’q qilish vazifa qilib belgilandi. Sun’iy yo’ldoshdan olingan cho’lning xaritasiga va bazaning planiga qarab, bazaning barcha mumkin bo’lgan joylashishlar sonini aniqlang.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylining 1-qatorida N_b va M_b sonlar, keyingi N_b ta qatorda bazaning plani(terrorchilar) berilgan bo’lib, ularning har birida M_b ta belgi – “#” belgisi(ASII 35) yoki , ” .” belgilari(ASCII 46) berilgan. “#” belgisi baza bo’lagini , ” .” esa qum ni bildiradi. Ulardan keyingi qatorda N_d va M_d sonlari berilgan. Keyingi N_d ta qatorning har birida M_d tadan –

“#” yoki ” .” belgilar joylashgan bo’lib, ular cho’lning terrorchilar joylashgan taxminiy xaritasini bildiradi.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida terrorchilar bazasining barcha mumkin bo’lgan joylashishlari sonini chiqaring.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	2 2 #. ## 3 5 #.#.# ##### .###.	4
2	1 3 #.. 3 6 ##..## .#.#.# #.#...	6
3	3 3 #.. #.# #.. 5 36 #.....#.....#... #.#....#.#...#.....#..... #.....#.#...##....#...##.....###.. .....#.....#...#.#...##.....#.#	4

	.....#.....#.....#..	

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n,i,j,m=0,m1=0,s=0,n1,k,l,g,t; String c=""; n=sc.nextInt(); m=sc.nextInt();
char b[][]=new char[n][m];
//n,m-terrorchilar bazasining o`lchamlari,b[][]-baza planidan iborat massiv
for(k=0;k<n;k++) { c=sc.next(); for(l=0;l<m;l++) { b[k][l]=(char)c.charAt(l); } }
n1=sc.nextInt(); m1=sc.nextInt(); char a[][]=new char[n1][m1];
//n1,m1-terrorchilar joylashgan hudud o`lchamlari, a[][] taxminiy xarita massivi
for(i=0;i<n1;i++) { c=sc.next(); for(j=0;j<m1;j++) { a[i][j]=(char)c.charAt(j); } }
 // a[][] da b[][] ko`rinishidagi baza plani borligini tekshirish
for(i=0;i<n1;i++) { if(i+n>n1) break;
for(j=0;j<m1;j++) { if(j+m>m1)break;
//←tekshirish davomida taxminiy xarita o`lchamidan chiqib ketmaslik
g=0;t=0; for(k=0;k<n;k++) {
for(l=0;l<m;l++) { if((b[k][l]=='#')) { t++; if(a[i+k][j+l]=='#') g++; } }
/*baza planidagi # mavjud bo`lgan nuqtada taxminiy xaritada topilgan baza planning ham shu nuqtasida # mavjudligini tekshirish*/ }
if(t==g) s++; //# lar soni teng bo`lsa, xaritada 1 ta baza plani mavjud } }
pw.println(s);
```

№ 413 Masala

Harbiy baza.

Sun’iy yo’ldoshdan raqibning maxfiy harbiy bazasi tasviri olindi. Bazadagi barcha qurilishlar muz bo’laklaridan yasalgan bo’lib, rasmda kvadrat shaklga ega va nolmas uzunlikdagi umumiy devorlarga ega emas. Turli qurilishlar devorlari rasmning chegaralariga paralelligini aniqlandi.

Bazada nechta bino(qurilish) borligini aniqlang.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylining 1- qatorida N va M - bazaning vertikal va gorizontal bo'yicha o'lchamlari berilgan. Keyingi N ta qatorda M tadan belgi bo'lib, " ." –bo'sh joyni, "#" – qurilish elementini bildiradi.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT faylida bazadagi qurilishlar soni chiqaring.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	8 6 ....##. ...##. ..... .###.. .###.. .###..	2

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int a=Integer.parseInt(sc.next()),b=Integer.parseInt(sc.next()),x,y,s=0,i,j,i1,j1;
char [][]m=new char[a][b]; //m[][] bazadagi elementlardan tuzilgan massiv
String s1; char k;
for(i=0;i<a;i++) { s1=sc.next(); for(j=0;j<b;j++) m[i][j]=s1.charAt(j); }
for(i=0;i<a;i++) for(j=0;j<b;j++) if(m[i][j]=='#') { s++; //bazani aniqlash
 x=i;y=j; //aniqlangan bazani yo`qotish→
 while(m[x][y]=='#') {
 while(m[x][y]=='#') { m[x][y]='.'; if(y<b-1) y++; } //satr bo`yicha tekshirish
 y=j; if(x<a-1) x++; } } //ustun bo`yicha tekshirish
 pw.print(s);
```

№ 198 Masala

Chiziqli tenglamalar sistemasi.

“Aynimagan”, N o’zgaruvchili N ta chiziqli tenglamaga ega sistemani yechish talab etiladi:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots \\ a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n = b_i \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases}$$

Kiritiluvchi ma`lumotlar: INPUT.TXT faylining 1-qatorda N-tenglamalar soni berilgan, keyingi N ta qatorning har biri (N+1) ta butun son: i- satrdagi tenglama– N ta a_{ij} va b_i koeffitsientlardan iborat (N<=100, $|a_{ij}|_i|<10^4$).

**Chiqariluvchi ma`lumotlar: OUTPUT.TXT** faylida yechimlarni bo`sh joy bilan ajratilgan holda chiqaring. Ular absolyut qiymati bo’yicha 10 dan oshmaydi.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	2 5 9 62 9 -3 54	7 3
2	3 7 -9 1 2 0 6 5 -15 3 -3 2 -9	8 5 -9

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
//dastur Gaus metodi asosida tuzilgan
int n=sc.nextInt(),i,j,l,h,u=0; double s=0,d=0,m=0; double a[][]=new double[n][n];
//noma`lum oldidagi koeffitsientlardan tuzilgan matritsa
double b[]=new double[n];//ozod hadlardan tuzilgan matritsa
double x[]=new double[n];//tenglama ildizlaridan tuzilgan massiv
```

```
for(i=0;i<n;i++) { v=0; j=0;
while(j<n){ s=Double.parseDouble(sc.next()); a[i][j]=s; j++; }
s=Double.parseDouble(sc.next()); b[i]=s; }
/*a[][] matritsa dioganalida nol elementlar mavjud bo`lsa shu satrni shu ustundagi elementi nol bo`lmagan boshqa satr(1-satrdan boshlab tekshirib ketiladi)ga almashtirish*/
for(l=0;l<n;l++) { if(a[l][l]==0) { for(j=0;j<n;j++) { if(a[j][l]!=0) {
for(i=0;i<n;i++) { m=a[j][i]; a[j][i]=a[l][i]; a[l][i]=m; }
m=b[l]; b[l]=b[j]; b[j]=m; } } } }
for(l=0;l<n;l++) { if(a[l][l]==0) { for(j=l;j<n;j++) { if(a[j][l]!=0) {
for(i=0;i<n;i++){ m=a[j][i]; a[j][i]=a[l][i]; a[l][i]=m; }
m=b[l]; b[l]=b[j]; b[j]=m; } } } }
for(l=0;l<n;l++) { d=a[l][l];//d-dioganaldagi element
//shu satrdagi elementlarni shu satrdagi dioganal elementiga bo`lish
for(h=l;h<n;h++) a[l][h]=a[l][h]*1.0/d; b[l]=b[l]*1.0/d;
//shu satrdan keyingi satrlardagi l- ustun elementlarini nolga aylantirish
for(i=l+1;i<n;i++) { d=a[i][l];
for(j=l;j<n;j++) { a[i][j]=-1*a[l][j]*d+a[i][j]; } b[i]=-1*b[l]*d+b[i]; } }
x[n-1]=b[n-1];//eng so`nggida n- tenglamadan xn nopma`lum topiladi
//xn yordamida qolgan elementlarni ham topamiz
for(l=n-2;l>=0;l--) { s=0; for(h=l+1;h<n;h++) s=s+a[l][h]*x[h]; x[l]=b[l]+(-1*s); }
for(l=0;l<n;l++) pw.print(Math.round(x[l])+" ");
```

2.4. Satrni tahrirlash(Разбор строк)

№ 44 Masala

Strelkalar

Faqat ‘>’, ‘<’ va ‘-‘ belgilaridan iborat ketma-ketlik berilgan. Berilgan ketma-ketlikda yashiringan strelkalar sonini qidirib toping. Strelka – bu ‘>>-->’ va ‘<--<<’ ko`rinishlardagi satr qismi.

**Kiritiluvchi ma`lumotlar: INPUT.TXT** faylida ‘>’, ‘<’ va ‘-‘ belgilardan iborat(bo`sh joylarsiz) satr berilgan. Satr uzunligi 250 belgidan dan oshmaydi.

Chiqariluvchi ma`lumotlar: OUTPUT.TXT fayliga qidirilayotgan strelkalar sonini chop eting.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	<<<<>>--><--<<--<<>>>--><<<<<	4

Qo`yilgan masalaning dasturi Java dasturlash tilida quyidagicha bo`ladi:

```
import java.io.*; import java.util.*;
class Main{ public static void main(String mm[])throws IOException{
Scanner sc=new Scanner(new File("input.txt"));
PrintWriter pw=new PrintWriter(new File("output.txt"));
String s=sc.next(); int n=s.length(),i,m1=0,m2=0,k=0;pw.println(n);
for(i=0;i<n;i++){ if(i!=n-1) {m1=(int)(s.charAt(i)); m2=(int)(s.charAt(i+1));}
if((m1==45)&(m2==45)&(i!=0)&(i!=n-1)) { if((int)(s.charAt(i-1))==60) {
if(((int)(s.charAt(i+2))==60)&((int)(s.charAt(i+3))==60)) k++;}}
if((m1==45)&(m2==45)&(i>1)&(i!=n-1)) {
if(((int)(s.charAt(i-1))==62)&((int)(s.charAt(i-2))==62)){
if((int)(s.charAt(i+2))==62) k++;}} }
/* ’-‘belgisining ASCII jadvalidagi kodi 45; ‘<’belgisining kodi esa 60; ’>’
belgisining kodi esa 62;*/
pw.println(k); pw.close(); }}
```

Keyingi masalalarning dasturini faqat asosiy qismini keltiramiz, ya`ni barcha masalalarga dastur tuzilishi quyidagicha bo`ladi. Faqat asosiy qism o`zgaradi.

```
import java.util.*; import java.io.*; import java.math.*;
class Main{ public static void main(String mm[])throws IOException{
 Scanner sc=new Scanner(new File("input.txt"));
 PrintWriter pw=new PrintWriter(new File("output.txt"));
```

Asosiy qism

sc.close();pw.close(); } }

№95 Masala

Numerolog

Numerolog inson taqdirini aytib berish uchun inson hayotining har bir sekundini oladi, undan so`ng hosil bo`lgan sondagi barcha raqamlarni qo`shadi. Agar hosil bo`lgan sonning raqamlari soni 1 dan ortiq bo`lsa, yana barcha raqamlar yig`indisi hisoblanadi, toki kelib chiqadigan sonda birta raqam qolguncha. Kelib chiqgan raqam va berilgan sonni raqamga aylantirish uchun bajarilgan amallar soni asosida numerolog inson taqdirini aytib beradi. Numerolog hisob-kitobni yoqtirmaydi, lekin u juda katta sonlar bilan ish ko`radi. Shunday dastur tuzingki, u numerologning barcha hisob-kitoblarini amalga oshirsin.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylida N – inson hayotining sekundlardagi ifodasi keltirilgan ($1 \le N \le 10^{1000}$).

**Chiqariluvchi ma`lumotlar:OUTPUT.TXT** faylida N sonidan hosil qilingan raqam va o`zgartiriishlar sonini bo`sh joy bilan ajratilgan holda chop eting.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	1	1 0
2	10	1 1
3	99	9 2

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
String n=sc.next(); //berilgan sonni n ga satr sifatida o`qib olish
long s=0,k=0,z=n.length();
/*z- berilgan satr uzunligi, k berilgan sonni necha qadamda raqamga aylanishi, s-berilgan sondan hosil bo`lgan raqam*/
int j=0;
```

```
if(z==1) { s=Long.parseLong(n); k=0; } //berilgan son raqam ekanligini tekshirish
else //berilgan son raqam bo`lmasa, uni raqamga aylantirish
while(j<z) { s=s+ (n.charAt(j)-'0'); if ((j==z-1)&& (s<10)) { k=k+1; break; }
if ((s>=10)&(j==z-1)) { n=""; n=n+s; z=n.length(); s=0; k=k+1; j=0; } else j++; }
pw.print(s+" "+k);
```

№163 Masala

5-sinflar uchun tenglama!

5-sinf o`quvchilari uchun tenglama 5 belgidan iborat satr ko`rinishida beriladi. Satrdagi ikkinchi belgi '+' (qo`shish) yoki '-' (ayirish), to`rtinchi belgi esa — '=' (tenglik) belgisidir. Birinchi , uchinchi va beshinchi belgilardan ikkitasi 0 dan 9 gacha bo`lgan raqamlardan va birtasi— x, noma`lumdan iborat.

x ni topishga imkon beruvchi dastur tuzing.

**Kiritiluvchi ma`lumotlar :**INPUT.TXT faylida yagona qatorda aytib o`tilgan tenglama keltirilgan.

Chiqariluvchi ma`lumotlar :OUTPUT.TXT faylida berilgan tenglamaning yechimi — x ni chop eting.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	x+5=7	2
2	3-x=9	-6

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
String s=sc.next();//s-berilgan tenglama
char x='+';//x-plus belgisi
 boolean t=false,t1=false;
/*t rost bo`lsa noma`lumning “=” belgisidan keyin joylashgan ,t1 rost bo`lganda
“+” belgisidan keyin joylashgan,ikkalasi ham yolg`on bo`lsa “+” belgisidan oldin
joylashgan bo`ladi*/
```

```
 n=s.length(), a=0,b=0,y=0;
 if(s.indexOf('x')==4) {
//x- tenglikdan o`ng tomonda joylashgan bo`lsa ,a,b-tenglama koeffitsientlari
 a=s.charAt(0)-'0';b=s.charAt(2)-'0'; x=s.charAt(1); t=true; }
 if(s.indexOf('x')==2) {
//x-“+” belgisidan keyin joylashgan bo`lsa
 a=s.charAt(0)-'0';b=s.charAt(4)-'0'; x=s.charAt(1); t1=true; }
 if(s.indexOf('x')==0) {
//x-“+”belgisidan oldin joylashgan bo`lsa
 a=s.charAt(2)-'0'; b=s.charAt(4)-'0'; x=s.charAt(1); }
 if(t)
 { if(x=='+') y=a+b; else y=a-b; }
 else { if(x=='+') y=b-a; else
 { if(t1) y=a-b; else y=b+a; } }
//y tenglamadagi noma`lum
 pw.print(y);
```

№231 Masala

Satrni yoyish

Lotin alifbosidagi bosh harflardan iborat satrni qaraylik. Masalan, AAAABCCCCCDDDD satrni olaylik. Bu satrning uzunligi 14 ga teng. Satrlar bir qancha lotin harflaridan iborat, takrorlangan belgilarni o`chirib va ularning o`rniga takrorlanishlar sonini yoziladi. Buning natijasida berilgan qator 4AB5C4D ko`rinishga keladi. Uning uzunligi esa 7 ga teng. Bu metod satrni yig`ish deb ataladi.

Yig`ilgan satrni boshlang`ich holatiga qaytaruvchi dastrur tuzing.

**Kiritiluvchi ma`lumotlar:**Kiritiluvchi INPUT.TXT faylida yig`ilgan satr berilgan. Satr nA va A ko`rinishdagi birlashmalardan iborat, bu yerda n — takrorlangan belgilar soni(2 dan 99 gacha bo`lgan butun sonlar), A esa — lotin

ailfbosining bosh harflari, agar birlashma ko`rinishi A ko`rinishida bo`lsa bu belgi takrorlanmagan belgi hisoblanadi. Satrning maksimal uzunligi 80 dan oshmaydi.

**Chiqariluvchi ma`lumotlar:** OUTPUT.TXT faylida boshlang`ich holatdagi satrni chop eting. Bu satrni namunadagidek 40 belgidan iborat satrlar jamlanmasi ko`rinishida chop eting(oxirgi satr uzunligi 40 dan kichik bo`lishi mumkin).

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	3A4B7D	AAABBBBDDDDDDD
2	22D7AC18FGD	DDDDDDDDDDDDDDDDDDDDDDDAAAAAAACFFFFFFFFFF FFFFFFFFGD
3	95AB	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAB
4	40AB39A	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA BAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
String s=sc.next(),z="",s1="";
/*s-berilgan satr,z- takrorlangan belgilar sonining satr ko`rinishi, s1 chiqariluvchi satr,n-s satr uzunligi*/
int n=s.length(),i,k=0,m,j;
for(i=0;i<n;i++){ m=s.charAt(i)-'0'; if((m<10)&(m>=0)&(i<n-1)) {
if((s.charAt(i+1)-'0'>=0)&(s.charAt(i+1)-'0'<=9))
{ z=z+m+(s.charAt(i+1)-'0'); i++; } else z=z+m; }
else { if(!z.equals("")) k=Integer.parseInt(z);
//demak qaralayotgan belgi k marta takrorlangan
else { if((s1.length()==40)) { pw.println(s1); s1=""; } s1=s1+s.charAt(i); } z="";
//qaralayotgan elementni s1 satrga k marta qo`shish
```

```
for(j=0;j<k;j++) { if((s1.length()==40)) { pw.println(s1);s1=""; } s1=s1+s.charAt(i);} k=0; } } pw.println(s1);
```

№13 Masala

Sigirlar va buzoqlar

Ali va Vali turli xil mantiqiy o`yinlar o`yinlar o`ynashmoqda. Yaqindagina Ali Valiga yangi «Sigirlar va buzoqlar» o`yinini o`rgatdi va endi ular kunda bu o`yinni o`ynaydilar. O`yinning maqsadi juda oddiy: Ali to`rt raqamdan iborat son o`ylaydi. Vali esa Ali o`ylagan bo`lishi mumkin bo`lgan turli sonlarni aytadi. Har gal Vali o`zining turli variantlarini aytganda Ali unga yordam beradi: sigirlar va buzoqlar sonini ma`lum qiladi, berilgan yodamdan keyin Vali o`ylangan sonni topishga bir necha bor urindi, toki topmagunicha. Sigirlar – bu Vali aytgan va Ali o`ylagan sonlarning ikkalasida ham mavjud va bir xil o`rinda joylashgan raqamlar. Buzoqlar – esa Vali aytgan va Ali o`ylagan sonlarning ikkalasida ham mavjud ammo bir xil o`rinda joylashmagan raqamlar soni. Masalan, agar Ali 5671 sonini o`ylagan bo`lsa, Vali esa 7251 variantni taklif qilgan bo`lsa, unda sigirlar soni 1 ta (faqat 1 ta 1 raqami ikkala sonda ham mavjud va o`z o`rnida joylashgan), buzoqlar soni esa 2 ta (faqat 7 va 5 o`z o`rnida joylashmagan). Ali kuchli matematik, ammo u ham adashishi mumkin. Ularga yordam bering, Ali o`ylagan va Vali taklif qilgan variantlardagi sigirlar va buzoqlar sonini topuvchi dastur tuzing.

**Kiritiluvchi ma`lumotlar:** INPUT.TXT faylida ikkita 4 xonali A va B natural sonlar bo`sh joy bilan ajratilgan holda berilgan, A – Ali o`ylagan son, B esa – Vali taklif qilgan son.

**Chiqariluvchi ma`lumotlar:** OUTPUT.TXT faylida ikkita butun son─ sigirlar va buzoqlar sonini bo`sh joy bilan ajratilgan holda chop eting.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	5671 7251	1 2
2	1234 1234	4 0

3	2034 6234	2 1

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
String n=sc.next(),m=sc.next(),n1="",m1="",m2="";
//n Ali o`ylagan son, m Vali aytgan son
int k=n.length(),k1=0,i,j,k2,s=0,b=0;
for(i=0;i<k;i++){ if(n.charAt(i)==m.charAt(i)) b++;
//b─n da ham m da ham uchragan va bir xil o`rinda turgan raqamlar miqdori
else { n1=n1+n.charAt(i);m1=m1+m.charAt(i); } } k1=n1.length(); k2=k1;
for(i=0;i<k1;i++){ for(j=0;j<k2;j++){ if(n1.charAt(i)==m1.charAt(j)) s++;
//s─ n da ham m da ham mavjud, lekin turli o`rinlarda turgan raqamlar miqdori
else m2=m2+m1.charAt(j); } k2=m2.length(); m1=m2; m2=""; }
pw.println(b+" "+s);
```

№164 Masala

Baxtli bilet - 2

Berilgan bilet baxtli bo`lishi uchun, uni shunday ikkita qismga ajaratish kerakki bu qismlardan iborat bo`lgan sonlarning raqamli ildizlari teng bo`lishi kerak. Sonning raqamli ildizini hisoblash uchun, uning raqamlarining yig`indisini hisoblaymiz, olingan natija 10 dan katta yoki teng bo`lsa, yana uning ham raqamlari yig`indisini hisoblaymiz va h.k… toki 0 dan 9 gacha bo`lgan son hosil bo`lgunga qadar davom ettiriladi, bu ─berilgan sonning raqamli ildizidir. Masalan, 0015420 nomerli bilet baxtli bilet hisoblanadi, chunki uni 0015 va 420 sonlardan iborat qismlarga ajratsak, bu sonlarning raqamli ildizlari tengdir.

Berigan biletning baxtli bo`lishini aniqlovchi dastur yuzing.

Kiritiluvchi ma`lumotlar: Kiritiluvchi INPUT.TXT faylida baxtli deb hisoblanayotgan biletning raqami berilgan. Bilet raqami 0 dan boshlanishi mumkin va 100 belgidan oshmaydi.

**Chiqariluvchi ma`lumotlar:**Agar bilet chindan ham baxtli bo`lsa «YES», aks holda «NO» so`zlarini OUTPUT.TXT fayliga chop eting.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	0015420	YES
2	00100	NO

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
String s=sc.next();//s─ baxtli deb hisoblanayotgan biletning raqami
int n=s.length(),s1,s2,j,i,k,m,z=0; boolean t=false;
for(i=0;i<n-1;i++){ j=0; k=i+1; s1=0; s2=0;
do { s1=s1+(s.charAt(j)-'0'); j++; } while(j<i+1);
//s1─ s satrning i-belgisigacha bo`lgan qismidagi raqamlar yig`indisini
if(s1==0) continue; do{ s2=s2+(s.charAt(k)-'0'); k++; } while(k<n);
//s2─s satrning i- belgisidan keyingi barcha raqamlari yig`indisini
while(s1>9){ z=0; do{ z=z+s1%10; s1=s1/10; } while(s1>0); s1=z; }
//s1 son 9 dan katta bo`lsa raqamga aylantirish
while(s2>9){ z=0; do{ z=z+s2%10; s2=s2/10; } while(s2>0); s2=z; }
//s2 son 9 dan katta bo`lsa raqamga aylantirish
if(s1==s2) { t=true; break; } }
//s1 va s2 ning tengligini tekshirish, shart bajarilsa jarayonni to`xtatish
if(t) pw.println("YES"); else pw.println("NO");
```

№322 Masala

So`z

Fibonachi sonlari quyidagi usulda hosil qilinadi: birinchi 2 ta fibonachi sonlari 1 ga teng, keyingilari esa undan oldingi 2 ta fibonachi soning yig`indisiga teng. Masalan, boshlang`ich 10 ta fibonachi soni quyidagiga teng : 1, 1, 2, 3, 5, 8, 13, 21, 34, 55. Berilgan satrdagi belgilar chapdan o`ngga qarab tartiblanadi va 1 dan boshlanadi.

Fibonachi sonlariga teng o`rinlarda turgan belgilardan tuzilgan satrni aniqlovchi dastur tuzing.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylida yagona satrda bosh lotin harflaridan iborat matn berilgan. Matndagi belgilar soni 30000 dan oshmaydi.

**Chiqariluvchi ma`lumotlar: OUTPUT.TXT** faylida Fibonachi sonlariga teng o`rinlarda turgan belgilardan tuzilgan satrni chop eting. Satrdagi belgilar tartibi berilgan matndagi kabi bo`lishi kerak.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	A	a
2	Abc	abc
3	Abcdefghij	abceh

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
String s=sc.next(),s1=""; int n=s.length(),x=1,y=1,i=0,z=0;
while(y<=n){ s1=s1+s.charAt(y-1); z=y; y=z+x; x=z; } pw.print(s1);
```

№168 Masala

Natural sonlar qatori

Hech qanday bo`sh joylarsiz ketma-ket yozilgan natural sonlardan iborat satr berilgan. Bu satr quyidagi ko`rinishda boshlanadi 123456789101112131415161718192021… .

Ushbu qatorda birinchi uchragan o`nli ko`rinishdagi N sonining joylashgan o`rnini aniqlovchi dastur tuzish kerak.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylida N soni berilgan ($1 <= N <= 10^4$).

**Chiqariluvchi ma`lumotlar: OUTPUT.TXT** yagona sonni – berilgan satrda birinchi uchragan N sonining joylashgan o`rnini chiqaring.Berilgan satrni o`rinlashtirish birdan boshlanadi.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	45	4
2	101	10
3	142	73

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
 int i=0,k=0,j=0,d=0; String c=sc.next(),s="1234";
/*s ning bunday olinishiga sabab cko`pi bilan 4 xonadan iborat bo`lishi mumkin,
c-berilgan natural son*/
j=4;
//s ga 4 va 4 gacha bo`lgan natural sonlar qo`silganligi uchun j 4 dan boshlanadi
while(j<=10000){ j++; i=s.indexOf(c); // i- c ning s satrdagi indeksi
if(i!=-1)break; d=d+s.length()-c.length()+1 ;
//d ga s satrning kesib tashlangan qismi uzunligini qo`shib boramiz
s=s.substring(s.length()-c.length()+1);
//s dan c ning uzunligidan 1 ta kamni qirqib tashlaymiz
s=s+j; } // va keyingi natural sonni qo`shamiz
if(i!=-1) pw.println(d+(i+1)); /*d gaqirqib tashlangandan keyingi s dagi c ning
indeksi i va 1ni qo`shamiz, indeks 0 dan boshlanganligi sababli*/
```

№600 Masala

Tilni anglash

Formal grammatikalar va avtomatikalarni asosiy tushunchasi formal til hisoblanadi. Noformal tarzda formal tilni simvollardan tashkil topgan satrlar to`plami sifatida ta`riflash mumkin.

Bu masalada berilgan so`zning$\{0^n1^n2^n,\ n \geq 1\}$ ushbu tilga tegishli ekanligini tekshirish kerak. Bu tilga quyidagi tuzilishga ega: nollar qancha bo`lsa shuncha birlar va birlar qancha bo`lsa shuncha ikkilardan iborat so`zlar kiradi. Bunda

ixtiyoriy nol ixtiyoriy birdan ko`ra so`z boshiga yaqin joylashgan, ixtiyoriy bir esa so`zda ixtiyoriy ikkidan ko`ra so`z boshiga yaqin joylashgan bo`ladi. Masalan, 001122 so`zi bu tilga tegishli, lekin 0000111122220 so`z esa bu tilga tegishli emas.

**Kiritiluvchi ma`lumotlar: INPUT.TXT** faylining birinchi qatorida musbat n ($n \leq 10$) – tekshiriladigan so`zlar soni. Keyingi n ta qatorning har birida birtadan so`z yozilgan. So`zlarning uzunligi 30 000 dan osmaydi va faqatgina nol, bir, ikkilardan tuzilgan. Har bir so`z hech bo`lmaganda birta belgidan iborat.

**Chiqariluvchi ma`lumotlar: OUTPUT.TXT** fayli n ta satrdan iborat bo`lishi kerak. Kiritiluvchi faylning har bir satriga mos holda ushbu satrdagi so`z berilgan tilga tegishli bo`lsa YES, aks holda NO – javoblarini chop eting.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	3 001122 00011122222 000111222	YES NO YES
2	2 0000111122220 012	NO YES

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n = 0,slen=0; int cnt0, cnt1, cnt2; boolean b0, b1, b2, res;
/*satr elementi 0,1,2 lardan biri bo`lsa,ularga muvofiq holda b0,b1,b2 lardan biri true qiymatni oladi,res til qoidasi bajarilganda true qiymatni oladi*/
n=sc.nextInt(); //n-so`zlar soni
String s;
for (int w = 0; w<n; w++) { cnt0 = 0; cnt1 = 0; cnt2 = 0;
b0 = true; b1 = false; b2 = false; res = true; s=sc.next(); slen=s.length();
```

```
//s-so`z, slen –so`z uzunligi
for (int i = 0; i<slen; i++){if (s.charAt(i) == '0') cnt0++;
if (s.charAt(i) == '1') cnt1++; if (s.charAt(i) == '2') cnt2++; }
if ((cnt0!=cnt1)||(cnt2!=cnt1)) res = false; else {//demak 0,1,2 lar soni teng
for (int i = 0; i<slen; i++){if (s.charAt(i)=='0') { if (b1||b2) { res = false; break; } }
//noldan oldin 1 yoki 2 mavjud emasligini tekshirish
if (s.charAt(i)=='1') { if (b2) { res = false; break; } }
//birdan oldin ikki mavjud emasligini tekshirish
if (s.charAt(i)=='0') b0 = true; if (s.charAt(i)=='1') b1 = true;
if (s.charAt(i)=='2') b2 = true; } }
if (res) pw.print("YES\n"); else pw.print("NO\n"); }
```

№287 Masala

Professor

Qaysidir mashhur universitetning mashhur professorlaridan biri maruzasini shunday tez o`qirdiki, bu ma`ruzadan biror nimani tushunib bo`lmasdi. Ammo yaqinda Bobur ismli bir talaba professor foydalanadigan so`zlar lug`atini tuzishga qaror qildi. Bu maqsadda u hatto bir leksiya qatnashib, ma`ruzani diktofonga yozib oldi. Undan keyin Bobur uyiga borib, yozib olingan leksiyani o`n marta sekinlashtirib tingladi va professor aytganlarining barini qog`ozga tushirdi.

Aammo bunda asosiy muammo professor juda tez gapirganligidan ma`ruzani sekinlashtirib eshitganda ham so`zlarni anglab bo`lmasdi. Shunday qilib boburda n ta kichik lotin harflaridan iborat ma`ruza matnlari bor. Endi bobur professor ma`ruzasida foydalangan m uzunligdagi so`zlar miqdori qanchaligini bilishni xoxlaydi.

**Kiritiluvchi ma`lumotlar:INPUT.TXT** faylining 1- qatorida n va m ($1 \le m \le n \le 100$) – professor ma`ruzasining uzunligi va so`zlar uzunligi berilgan. Ikkinchi qatorda esa n ta lotin belgilaridan iborat satr- professor ma`ruzasi berilgan.

**Chiqariluvchi ma`lumotlar:OUTPUT.TXT** fayliga yagona son professor ma`ruzasida foydalangan so`zlar miqdorini chiqaring.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	3 1 abc	3
2	10 3 bbaabbbabb	6

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n=sc.nextInt(),m=sc.nextInt(),i,j,j1=0,k=0;
// n– professor leksiyasining uzunligi va m–so`zlar uzunligi
String s=sc.next(),a[]=new String[100],z;//s professor ma`ruzasi
a[0]=s.substring(0,m);
for(i=0;i<=n-m;i++){ if(i!=n-m) z=s.substring(i,i+m); else z=s.substring(i);
//i-belgidan boshlab m tadan qirqib olish
j=j1; while(j>=0) { if(z.equals(a[j])) k++; j--; }
//massivda shunday so`z borligini tekshirish
if(k==0) { j1++;a[j1]=z; }
//agar bo`lmasa massivga qo`shib olish va so`zlar sonini j1 ga hisoblab borish
else { k=0; continue; } } pw.println(j1+1);
```

2.5.Ketma-ketrliklar va ularni saralash(Сортировка и последовательности)

№119 Masala

Vaqtni tartiblash

Berilgan vaqt momentini soat, minut, sekund bo`yicha tartiblash

Kiritiluvchi ma`lumotlar: 1- qatorda N (1<=N<=100)- kiritiladigan vaqt momentllari soni, keyingi N qatorda kiritiladigan vaqt momentlari berilgan. Har bir

vaqt momenti 3 ta butun sondan iborat –soat(0dan 23 gacha), minut(0 dan 59 gacha) va sekund(0 dan 59 gacha).

**Chiqariluvchi ma`lumotlar:** Vaqt momentlarini ortiqcha nollarsiz o`suvchi tartibda chiqaring.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	4 10 20 30 7 30 00 23 59 59 13 30 30	7 30 0 10 20 30 13 30 30 23 59 59

Qo`yilgan masalaning dasturi Java dasturlash tilida quyidagicha bo`ladi:

```
import java.util.*; import java.math.*; import java.io.*;
class Main{ public static void main(String mm[])throws IOException {
Scanner sc=new Scanner(new File("input.txt"));
PrintWriter pw=new PrintWriter(new File("output.txt"));
int n=sc.nextInt(),m[][]=new int[n][3],i,j,b=0;
for(i=0;i<n;i++){ for(j=0;j<3;j++){ m[i][j]=sc.nextInt(); } }
for(j=0;j<n-1;j++){ for(i=0;i<n-1;i++){
if((m[i][0]>m[i+1][0])||((m[i][0]==m[i+1][0])&(m[i][1]>m[i+1][1]))
||((m[i][0]==m[i+1][0])&(m[i][1]==m[i+1][1])&(m[i][2]>m[i+1][2]))){ b=m[i][0];
m[i][0]=m[i+1][0]; m[i+1][0]=b; b=m[i][1]; m[i][1]=m[i+1][1];
m[i+1][1]=b; b=m[i][2]; m[i][2]=m[i+1][2]; m[i+1][2]=b; } } }
for(i=0;i<n;i++) pw.println(m[i][0]+" "+m[i][1]+" "+m[i][2]); pw.close(); } }
```

Keyingi masalalarning dasturini faqat asosiy qismini keltiramiz, ya`ni barcha masalalarda dastur tuzilishi quyidagicha bo`ladi. Faqat asosiy qism o`zgaradi.

```
import java.util.*; import java.io.*; import java.math.*;
class Main{ public static void main(String mm[])throws IOException{
```

```
Scanner sc=new Scanner(new File("input.txt"));
PrintWriter pw=new PrintWriter(new File("output.txt"));
```

Asosiy qism

```
sc.close();pw.close(); } }
```

№5 Masala

Statistika

Vali ingliz tilini yoqtirmaydi, jazolanmaslik uchun har chorakda hech bo`lmaganda 4 olishga urunadi. Keyingi chorakda Vali quyidagi qoidaga e`tibor qildi: oyning toq kunlarida 3, juft kunlarida esa 4 baho oladi. U qaysi kunlari bu baholarni olganligini eslaydi. U qancha 3 va qancha 4 baho olganligini bilish uchun bu kunlarni yozib oladi. Valiga toq va juft sanalarni turli qatorlarga joylashtirishga yordam bering. Uning hisobiga ko`ra 4 baholar 3 baholardan kam bo`lmasligi kerak.

**Kiritiluvchi ma`lumotlar: INPUT.TXT** faylining 1-qatorda yagona N— butun sonli massiv elementlari soni keltirilgan (1<=N<=100). Keyingi qatorda ushbu massiv elementlari keltirilgan. Har bir element 1 dan 31 gacha bo`lgan natural son. Massivning barcha elementlari bo`sh joy bilan ajratilgan.

**Chiqariluvchi ma`lumotlar: OUTPUT.TXT** faylining 1-qatorda Vali 3 olgan kunlar, 2- qatorda esa 4 olgan kunlari ro`yxati berilgan, kunlar bo`sh joy bilan ajratilgan. Keyingi qatorda agar Vali 4 baho olsa “YES”, aks holda “NO” so`zlari chiqarilishi kerak.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	5 4 16 19 31 2	19 31 4 16 2 YES
2	8 29 4 7 12 15 17 24 1	29 7 15 17 1 4 12 24

		NO

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
String s="",s1="",s2="";
int n=sc.nextInt(),a[]=new int[n],i,j=0,k=0,b1[]=new int[n],b2[]=new int[n];
/n Vali baho olgan kunlar soni
for(i=0;i<n;i++) a[i]=sc.nextInt(); //a[] Vali baholangan kunlardan iborat massiv
for(i=0;i<n;i++) { if(a[i]%2!=0) {b1[j]=a[i];j++; } else {b2[k]=a[i];k++;} }
/*Vali 3 olgan kunlar soni j va shu kunlardan iborat massiv-b1[], Vali 4 olgan kunlar soni k va shu kunlardan iborat massiv-b2[].*/
for(i=0;i<j;i++) s1=s1+b1[i]+" "; if(s1.length()>0) s1=s1+"\n";
for(i=0;i<k;i++) s2=s2+b2[i]+" "; if(s2.length()>0) s2=s2+"\n";
s=s+s1+s2; if(k>=j) s=s+"YES"; else s=s+"NO";pw.println(s);
```

№9 Masala

Uyga vazifa

Petya sinfdoshlaridan ko`ra matematikani yaxshi bilardi, shuning uchun o`qituvchisi unga qiyinroq vazifa berdi: berilgan sonlar ketma-ketligidagi barcha musbat sonlar yig`indisi hamda shu ketma-ketlikdagi min va max elementlar oralig`idagi elementlar ko`paytmasini topish kerak. Bunda min va max elementlar ushbu ketma-ketlikda faqat bir marta uchraydi deb hisoblanadi. O`qituvchisi Petyaga bir qancha shunday masalar berdi, ammo Petya kuchli dasturchi bo`lganligi sababli ushbu masalalarni yechuvchi dastur tuzdi. Siz ham shunday dastur tuza olasizmi?

**Kiritiluvchi ma`lumotlar: INPUT.TXT** faylining birinchi qatorida N – massiv elementlari soni keltirilgan. Ikkinchi qatorda N ta son – massiv elementlari berilgan. Barcha elementlar bo`sh joy bilan ajratilgan. Kiritiluvchi fyaldagi har bir son absolyut qiymati bo`yicha 10^2 dan oshmaydi.

**Chiqariluvchi ma`lumotlar: OUTPUT.TXT** faylida yagona qatorda bo`sh joy bilan ajratilgan holda ikkita son: musbat elementlar yig`indisi hamda min va

max elementlar oralig`idagi elementlar ko`paytmasini chiqaring. Yig`indi va ko`paytma moduli bo`yicha $3*10^4$ dan oshmaydi.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	5 -7 5 -1 3 9	17 -15
2	8 3 14 -9 4 -5 1 -12 4	26 180
3	10 -5 1 2 3 4 5 6 7 8 -3	36 5040

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n=sc.nextInt(),a[]=new int[n],min=0,max=0,s=0,p=1,i;
//n massivdagi elementlar soni,a[] esa massivning o`zi
for(i=0;i<n;i++){ a[i]=sc.nextInt(); if(a[i]>0) s+=a[i];
if(a[i]<a[min]) min=i; if(a[i]>a[max]) max=i; }
/*s-massiv elementlari yig`indisi, a[min]-massivning minimal elementi, a[max]-massivning maksimal elementi */
if(min>max){ n=min; min=max; max=n; } for(i=min+1;i<max;i++) p*=a[i];
//p a[] massivning min va max elementlari oralig`idagi elementlar ko`paytmasi
pw.print(s+" "+p);
```

№64 Masala

Tub sonlar qatori

Natural son(1 dan katta) tub son deb atalishi uchun uning 1 va o`zidan boshqa bo`luvchilari bo`lmasligi kerak. Masalan 2,3,5,7,11 va23 – tub sonlar. Raqamlar ketma-ketligi tub sonlar qatori deb atalishi uchun 2 dan boshlab barcha tub sonlar o`suvchi tartibda ketma-ket yozilishi kerak. Ushbu qator boshlanishi quyidagi

ko`rinishda beriladi: 23571113171923… Tub sonlar qatoridagi ko`rsatilgan o`rinda turgan belgi(raqam)ni topish kerak. Belgilar 1 dan boshlab o`rinlashtiriladi.

**Kiritiluvchi ma`lumotlar: INPUT.TXT** faylining 1-qatorda berilgan o`rinlar soni – M, 2-qatorda M ta son bo`sh joy bilan ajratilgan holda berilgan. Har bir son – tub sonlar qatoridagi belgi o`rnini ifodalaydi. (1 ≤ M ≤ 1000, 1 ≤belgining joylashgan o`rni ≤ 10000).

**Chiqariluvchi ma`lumotlar: OUTPUT.TXT** fayliga ko`rsatilgan o`rinda turgan M ta belgi(raqam) dan tuzilgan satr (belgilar ketma-ket bo`sh joy bilan ajratilmagan)ni chiqaring.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	3 1 4 11	271
2	5 2 5 6 8 12	31139

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n=sc.nextInt(),a[]=new int[n],k=0,i,b=0,j;
//n berilgan o`rinlar soni, a[] berilgan o`rinlardan iborat massiv
long d; String s="",z="";
for(i=0;i<n;i++) { a[i]=sc.nextInt(); if(a[i]>k) k=a[i];} i=2;
while((s.length()<=k)){ d=Math.round(Math.sqrt(i));
if((i!=2)&(i%2!=0)&(i%d!=0)) for (j=1;j<=d;j++) if (i%j==0) { b=b+1;}
if(i==2) b=1; if(b==1) s=s+i; b=0; i++; }
//s-tub sonlardan iborat, a[] massivdagi eng katta elementga teng uzunlikdagi satr
for(i=0;i<n;i++) z=z+s.charAt(a[i]-1);
//z- s satrdagi berilgan o`rinlarda joylashgan belgilardan tuzilgan satr
pw.println(z);
```

№482 Masala

Qisqa ketma-ketlik

N- butun son berilgan. $S_1S_2S_3...S_k...$-ketma-ketlik “qisqa ketma-ketlik“ deb ataladi va undagi har bir S_k —1 dan k gacha bo`lgan raqamlar ketma-ketligidan iborat. Masalan ketma-ketlikning boshlang`ich 75 elementi quyidagicha:

112123123412345123456123456712345678123456789123456789101234567891011123456.

Qisqa ketma-ketlikdagi n- belgisini aniqlovchi dastur tuzing.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylida yagona satrda N soni berilgan(0<N<32768).

Chiqariluvchi ma`lumotlar: OUTPUT.TXT fayliga ketma-ketlikdagi n-belgini chiqaring.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	3	2
2	20	5

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n=sc.nextInt(),i; //berilgan sonni o`qib olish
String s="",z=""; for(i=1;z.length()<=n;i++) { s=s+i; z=z+s; }
//s- masalada keltirilgan Sk , z- “qisqa ketma-ketlik “
 pw.println(z.charAt(n-1));
```

№56 Masala

Jonli Jurnal

Dasturchi Ali jonli jurnal blok-xizmatini yaratishda ishtirok etyapti. Rejalashtirilishicha bu xizmat “LiveJournal'a”ga qaraganda juda ko`p imkoniyatlarni taqdim etadi.

Ayni vaqtdagi muammo “LiveJournal'a”ning barcha bazaviy imkoniyatlarini amalga oshirishdir. Shunday imkoniyatlardan biri foydalanuvchi uchun do`stlar ro`yxatini hosil qilishdir. Ma`lumotlar: biror foydalanuvchining do`stlari ro`yxati va shu foydalanuvchi mavjud bo`lgan do`stlar ro`yxati berilgan.

Qaralayotgan foydalanuvchining do`stlari ro`yxati (Friends), o`zaro do`stlar ro`yxati(Mutual Friends) va do`stlari ro`yxatida mavjud bo`lmagan foydalanuvchilar ro`yxatini hosil qiling(Also Friend of).

**Kiritiluvchi ma`lumotlar:INPUT.TXT** faylining birinchi qatorida n($0 \leq n \leq 200$) qaralayotgan foydalanuvchining do`stlari soni, keyingi n qatorning har birida shu do`stlarning nomi, (n+2)-qatorda esa m ($0 \leq m \leq 200$) foydalanuvchi joylashgan do`stlar ro`yxatidagi foydalanuvchilar soni va keyingi m qatorda do`stlar ro`yxatidagi foydalanuvchilar nomlari keltirilgan.

Foydalanuvchi nomidan iborat satr uzunligi 20 dan oshmaydi, hamda faqat lotin alifbosidagi harflar va tire(“-“) belgisidan iborat. Har bir foydalanuvchi berilgan ro`yxatlarda faqat bir martadan uchrashi mumkin.

**Chiqariluvchi ma`lumotlar:OUTPUT.TXT** faylida qaralayotgan foydalanuvchining do`stlari ro`yxati(Friends), o`zaro do`st foydalanuvchilar ro`yxati(Mutual Friends) va do`stari ro`yxatida mavjud bo`lmagan foydalanuvchilar ro`yxatini(Also Friend of) chiqaring. Har bir ro`yxat alifbo bo`yicha tartiblangan va namunada ko`rsatilgan formatda chiqarilishi kerak.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	3 vasya-pupkin bill-hates ivan-ivanov 2 vasya-pupkin	Friends: bill-hates, ivan-ivanov, vasya-pupkin Mutual Friends: vasya-pupkin Also Friend of: destroyer

	destroyer	
2	0 0	Friends: Mutual Friends: Also Friend of:

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n=sc.nextInt(),j,i,k=0,m,l=0,z=0; String f[]=new String[n];
for(j=0;j<n;j++) f[j]=sc.next();
// f [] foydalanuvchining do`stlari ro`yxatidan iborat massiv
Arrays.sort(f); m=sc.nextInt(); String M[]=new String[n+m],A[]=new String[m];
String a[]=new String[m],s="";
for(j=0;j<m;j++) a[j]=sc.next();
// a [] foydalanuvchi joylashgan do`stlar ro`yxatidan iborat massiv
Arrays.sort(a);
for(i=0;i<n;i++){ for(j=0;j<m;j++) if(a[j].equals(f[i])){ M[z]=f[i];z++;a[j]="0"; } }
//M[] Mutual Friends massivi
for(j=0;j<m;j++) if(!a[j].equals("0")) { A[k]=a[j]; k++;}
//A[] “Also Friend of ” massivi
s=s+"Friends:"; for(i=0;i<n;i++) s=s+" "+f[i]+","; l=s.length();
if(s.charAt(l-1)==',') s=s.substring(0,l-1); s=s+"\nMutual Friends:";
for(i=0;i<z;i++) s=s+" "+M[i]+","; l=s.length();
if(s.charAt(l-1)==',') s=s.substring(0,l-1); s=s+"\nAlso Friend of:";
for(i=0;i<k;i++) s=s+" "+A[i]+","; l=s.length();
if(s.charAt(l-1)==',') s=s.substring(0,l-1); pw.print(s);
```

№91 Masala

Ikki ketma-ketlik

a_n va b_n ketma-ketliklar quyidagi ko`rinishda aniqlanadi: $a_1 = 2$, $a_2 = 3$, $a_3 = 4$, $a_4 = 7$, $a_n = b_{n-1} + b_{n-3}$, $n > 4$, $b_n$ — sonlar ketma-ketligi, $a_n$ ga kirmaydigan sonlardan iborat va o`suvchi tartibda yozilgan.

Shunday qilib a_n ketma-ketlik quyidagi ko`rinishda bo`ladi: 2, 3, 4, 7, 13, 15,..., b_n ketma-ketlik esa– 1, 5, 6, 8, 9, 10,....

Sizning vazifangiz a_n va b_n ni qidirib topshdan iborat.

Kiritiluvchi ma`lumotlar: INPUT.TXT faylida yagona satrda n- butun soni berilgan($1 \le n \le 10000$).

**Chiqariluvchi ma`lumotlar: OUTPUT.TXT** faylining 1- qatori $a_n$, 2- qatori esa – $b_n$ dan iborat bo`lishi kerak.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	4	7 8
2	10	25 16
3	6578	19731 9868

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n=sc.nextInt(),i,j,x,y,z,k; ArrayList a=new ArrayList(),b=new ArrayList();
a.add(2);a.add(3);a.add(4);a.add(7);
// a ketma-ketlikning birinchi 4 elemanti mavjud bo`lishi kerak
b.add(1);b.add(5);b.add(6);b.add(8);b.add(9);b.add(10);b.add(11);b.add(12);
/* b ketma-ketlik esa a[4] gacha bo`lgan natural sonlarning a ketma-ketlikda mavjud bo`lmaganlaridan iborat */
float m=0;
for(i=4;i<=n;i++){ x=(Integer)b.get(i-1); y=(Integer)b.get(i-3); a.add(x+y);
```

// a_n = x+y, y= b_{n-3} , x= b_{n-1}

x=(Integer)a.get(i-1); y=(Integer)a.get(i);

if ((i>=5)&(b.size()<n)) for(j=x+1;j<y;j++) b.add(j); }

//a_{n-1} va a_n oralig`idagi joylashgan natural sonlarni b ga o`zlashtirish

pw.println(a.get(n-1)+"\n"+b.get(n-1));

№326 Masala

Ketma-ketlikni o`zgartirish – 2

N ta butun sondan iborat kema-ketlik berilgan. Shunday sonni topingki u berilgan ketma-ketlikda eng ko`p marta uchrasin, agar bunday sonlar ko`p bo`lsa, ulardan eng kichigini qidirib toping va undan so`ng ushbu sonlarni berilgan ketma-ketlikning oxiriga joylashtiring. Qolgan sonlarning joylashgan o`rni o`zgarmaydi.

Masalan: 1, 2, 3, 2, 3, 1, 2 ketma-ketlik o`zgartirilgandan keyin 1, 3, 3, 1, 2, 2, 2 ketma-ketlikga aylanadi.

Berilgan masalani yechuvchi dastur tuzing.

**Kiritiluvchi ma`lumotlar:**INPUT.TXT faylning 1-qatori n — kiritiluvchi kema-ketlikning elementlari sonidan iborat($3 \le n \le 100$). Keyingi qatorda n ta butun sondan iborat kiritiluvchi ketma-ketlik joylashgan, elementlar moduli bo`yicha 100 dan oshmaydi. Qatordagi barcha raqamlar bo`sh joy bilan ajratilgan.

**Chiqariluvchi ma`lumotlar:**OUTPUT.TXT fayliga o`zgartirishdan keyingi ketma-ketlikni chiqaring. Ketma-ketlikdagi barcha raqamlar bo`sh joy bilan ajratilgan bo`lishi kerak.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	7 1 2 3 2 3 1 2	1 3 3 1 2 2 2

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n=sc.nextInt(),i,j,c=0;//n- kiritiluvchi kema-ketlikning elementlari soni
int a[]=new int[n],b[]=new int[n],b1[]=new int[n],k=0,k1=0;
String s=""; for(i=0;i<n;i++) { a[i]=sc.nextInt();b1[i]=a[i];}
//a[], b1[] - kiritiluvchi kema-ketlikning elementlaridan tashkil topgan massiv
int min,max=0; Arrays.sort(b1);min=b1[0];
for(i=0;i<n-1;i++) { c=1; k1=i;
while (i<n-1) { if(b1[i]==b1[i+1]) c++;else { if(i>k1) i--; break;} i++;}
if((c>max)){ max=c; min=b1[i]; } if(c==max) if(b1[i]<min) min=b1[i]; }
//min-ketma-ketlikdagi eng ko`p uchragan son
i=0; while(i<n){ if(a[i]==min) a[i]=110; i++; }
//min ga teng elementlarni 110 bilan almashtirdik, sabab | a[i] |<=100
for(i=0;i<n;i++) if(a[i]!=110) { b[k]=a[i]; k++; }
/*shu elementlardan boshqa elementlarni b[] massivga o`zlashtiramiz va qolgan
qismini min bilan to`ldiramiz*/
for(i=k;i<n;i++) b[i]=min; for(i=0;i<n;i++) pw.print(b[i]+" ");
```

№17 Masala

G`aroyib maydoncha(pole chudes)

G`aroyib maydoncha o`yini uchun sektorlarga bo`lingan va strelkali aylana barabandan foydalaniladi. Har bir sektorga bir qancha raqamlar yozilgan. Turli sektorlardagi sonlar bir-biridan farq qilishi yoki farq qilmasligi mumkin. Bir safar boshlovchi o`yin qoidasini o`zgartirdi. Uning o`zi barabanni aylantiradi va o`yinda(bunda baraban ko`rinmaydi) barcha raqamlar baraban aylanishi jarayonida strelka ko`rsatgan tartibda joylashtiriladi. Shundan keyin boshlovchi o`yinchiga savol berdi: barabandagi sektorlarning minimal miqdori qancha bo`lishi mumkin? Boshlovchi savolining javobini topuvchi dastur tuzing.

**Kiritiluvchi ma`lumotlar:** INPUT.TXT faylining birinchi qatorida N – boshlovchi aytgan sonlar miqdori ($2 \le N \le 30000$). Ikkinchi qatorda shu N ta sonning o`zi, ya`ni barabanni aylantirish jarayonida strelka ko`rsatgan sonlar. Birinchi son har doim oxirgi son bilan bir xil bo`ladi(oxirida strelka yana birinchi

sektorga borib to`xtaydi). Barabandagi sektorlarga yozilgan sonlar– natural va 32000 dan oshmaydi.

**Chiqariluvchi ma`lumotlar:** OUTPUT.TXT faylida yagona son – barabanda mavjud bo`lgan minimal sektorlar sonini chop eting.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	13 5 3 1 3 5 2 5 3 1 3 5 2 5	6
2	4 1 1 1 1	1
3	4 1 2 3 1	3

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n=Integer.parseInt(sc.next()),list[]=new int[n];
//n –boshlovchi aytgan sonlar miqdori
for (int i = 0; i<n; i++) list[i]=Integer.parseInt(sc.next());
//list[] barabanni aylantirish jarayonida strelka ko`rsatgan sonlardan iborat massiv
n--; //oxirgi sektor birinchi sektor bilan bir xil bo`lganligi tufayli n 1taga kamayadi
int r=n,l,j; boolean t;//r-barabandagi sektorlar soni
for (int i=1;i<n;i++){ if (n%i==0){ l=i;j=0;
while ((list[j%l] == list [j]) && (j<n))j++;
//agar barabandagi sektorlar soni l ta bo`lsa u holda j n ga teng bo`ladi
if (j==n) t=true; else t=false; if (t) { r=i; break; } } } pw.println(r);
```

№32 Masala

Yillik hisobot

“Shox va Tuyoq” idorasida yillik hisobotni tayorlash vaqti kelgan edi. Hisobchiga kelgan ma`lumotlarga ko`ra bitim hujjatlarida chiqim a so`mni, kirim

esa b so`mni tashkil etadi. Ish jarayonida bu raqamlar to`g`riligining umuman ahamyati yo`q. Hisobchi o`zining quyidagi g`oyasini amalga oshirishga qaror qildi. Shunisi aniqki kompyuterga bu raqamlar betartib ravishda kiritilgan.Bundan hisobchi quyidagicha foydalanmoqchi: a va b sonlarning raqamlarini shunday o`zgartirish kerakki, natijada ularning ayirmasi maksimal bo`lsin va biror hodisa bo`lganda ham buni kotibaning xatosi deb hisoblashadi. Bunda kotibaning ushbu sonlardagi raqamlarni unutmasligi va nol birinchi raqam bo`la olmasligini hisobga olish kerak. Hisobchiga yordamlashuvchi dasrtur tuzing.

Kiritiluvchi ma`lumotlar: INPUT.TXT fayliga a va b sonlar kirtiladi ($-10^9 < a,b < 10^9$).

**Chiqariluvchi ma`lumotlar:** OUTPUT.TXT faylida ikki sonning maksimal farqini chiqaring, bunda birinchi son a sonining raqamlarini o`rin almashtirishdan hosil bo`lgan son, ikkinchi son esa b sonining raqamlarini o`rin almashtirshdan hosil bo`lgan son.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	18 10	71
2	1 -23	33

Bu masalaning dasturi boshqalariga o`xshamaganligi sababli to`liq holda keltirildi:

```
import java.util.*; import java.io.*;
 class yordam{//Main sinfiga yordamchi sinf
/*yuborilgan sonning raqamlarini o`rin almashtirish orqali undan maksimal sonni
hosil qilish funksiyasi→ */
public int max(int a){ int S[]=new int[20],A=a; int i=0,j,s,n,k,m=0;
boolean t; while(A>0){ S[i]=A%10; A=A/10; i++; }
```

```
//S[] massivga yuborilgan sonning raqamlarini o`zlashtirish
n=i; do{ t=false; for (i=0;i<=n-1;i++) for (j=0;j<=n-2;j++)
if(S[j]<S[j+1]){ s=S[j]; S[j]=S[j+1]; S[j+1]=s;t=true; } }//S[] massivni tartiblash
while(t);
k=1; for(i=n-1;i>=0;i--){ m=m+k*S[i];k*=10;} //massivni songa aylantirish
return m; }
/*yuborilgan sonning raqamlarini o`rin almashtirish orqali undan min sonni hosil
qilish funksiyasi*/
public int min(int b){
int S[]=new int[20],B=b; int i=0,j, s,m=0,n,k,z; boolean t;
while(B>0){ S[i]=B%10; B=B/10; i++ ; }
//S[] massivga yuborilgan sonning raqamlarini o`zlashtirish
n=i;
do{ t=false; for (i=0;i<n;i++)
for(j=0;j<n-1;j++) if(S[j]>S[j+1]){s=S[j];S[j]=S[j+1];S[j+1]=s;t=true;} }
while(t); //S[] massivni tartiblash
i=0; if(n>1) while(S[i]==0) {i++;} z=i;//z S[] massivdagi nollar soni
i=0; if(z>0) {S[i]=S[z+i]; S[z+i]=0; }
//z noldan katta bo`lsa u holda nollarni ulardan keyingi z ta son bilan almashtirish
k=1; for(i=n-1;i>=0;i--){ m=m+k*S[i]; k*=10;} //massivni songa aylantirish
return m; } }
class Main{
public static void main(String mm[])throws IOException{
Scanner sc=new Scanner(new File("input.txt"));
PrintWriter pw=new PrintWriter(new File("output.txt"));
int a=sc.nextInt(),b=sc.nextInt(),a1,b1;yordam k=new yordam();
//a,b berilgan sonlar
if(a>0) a1=k.max(a); else a1=-1*k.min(-1*a);
if(b>0) b1=k.min(b); else b1=-1*k.max(-1*b);
```

pw.print(a1-b1); pw.close(); } }

№224 Masala

Eng katta ko`paytma

N ta butun sondan iborat sonlar ketma-ketligi berilgan. Ketma-ketlikdan shunday 3 ta sonni tanlash kerakki ko`paytma maksimal bo`lsin.

Ushbu ko`paytmani aniqlovchi dastur tuzing.

**Kiritiluvchi ma`lumotlar:INPUT.TXT** faylining 1-qatorida berilgan sonlar ketma-ketligidagi elementlar soni N-keltirilgan ($3 \le N \le 10^6$). Keyingi qatorda shu ketma-ketlikning o`zi: N ta butun son berilgan, ular moduli bo`yicha 30000 dan oshmaydi.

**Chiqariluvchi ma`lumotlar:OUTPUT.TXT** fayliga ko`paytmani chiqaring.

Masalan

№	INPUT.TXT	OUTPUT.TXT
1	9 3 5 1 7 9 0 9 -3 10	810
2	3 -5 -30000 -12	-1800000

Qo`yilgan masala dasturining Java dasturlash tilida asosiy qismi quyidagicha bo`ladi:

```
int n=sc.nextInt(),i,k=0,z,a[]=new int[n];
/*n-ketma-ketlikdagi elementlar soni;a[]-ketma-ketlik elementlaridan tuzilgan massiv*/
long p=1; //k a[] dagi manfiy sonlar miqdori
for(i=0;i<n;i++) { a[i]=Integer.parseInt(sc.next()); if(a[i]<0) k++; }
Arrays.sort(a); //a[] ni tartiblash
if(n==3) p=(long)a[0]*a[1]*a[2]; //a[] ning elementlari 3 ta bo`lgan hol
else { if((k>1)&(k!=n)) {
//manfiy elementlar birtadan ko`p bo`lgan hol, n ga teng emas
```

```
if(a[0]*a[1]>a[n-2]*a[n-3]) p=(long)a[0]*a[1]*a[n-1];
else p=(long)a[n-2]*a[n-3]*a[n-1]; }
else p=(long)a[n-1]*a[n-2]*a[n-3]; }pw.println(p);
```

II bobning qisqacha xulosasi.

II bob “Mavzulashtirilgan olimpiada masalalari“ deb nomlanib, unda acmp.ru saytidagi mavzulashtirilgan olimpiada masalalaridan 5 ta mavzudagi masalalar keltirilgan. Bu bob 5 ta bo’limdan iborat bo`lib, har bir bo`limda bir mavzuga doir masalalarning o`zbek tilidagi shartlari va java dasturlash tilidagi dasturlari izohlari bilan berilgan.

Quyidagi mavzudagi masalalar keltirilgan:

1.Oddiy matematika(Простая математика).

2. Uzun sonlar arifmetikasi (Длинная арифметика).

3. Ikki o`lchovli massivlar (Двумерные_массивы).

4. Satrni tahrirlash(Разбор строк).

5. Ketma-ketrliklar va ularni saralash(Сортировка и последовательности).

Ushbu mavzularga doir keltirilgan barcha masalalar acmp.ru saytida tekshirilgan va barcha testlardan o`tgan.

Xotima

Ushbu bitiruv malakaviy ishimning mavzusi “Dasturlash bo`yicha olimpiada masalalari va ularni yechishga doir uslubiy ko`rsatmalar.(2-qism)” deb nomlanib, u kirish qismi, ikkita bob, boblar xulosalari, xotima va foydalanilgan adabiyotlardan iborat.

Bitiruv malakaviy ishining I bobi internet tizimida olimpiada masalalari va ularni yechishga bag’ishlangan bo`lib, 2 ta bo`limdan iborat.

1-bo`lim “Dasturlash bo’yicha olimpiadaga oid saytlar” deb nomlanib, internet tarmog’idagi dasturlash bo’yicha olimpiadaga oid turli xil saytlar, hamda ushbu saytlardan acmp.ru sayti va saytdagi masalalarni ko`rish, yozilgan dasturlarning to`g`riligini tekshirish imkoniga ega bo`lish uchun bajarilishi kerak bo`lgan ishlar ketma-ketligi keltirildi.

2-bo`lim “Java dasturlash tilida olimpiada masalallarini yechish va tekshirish” deb nomlanadi. Bu bo`limda olimpiada masalalarini yechish uchun java dasturlash tilidan foydalaniladigan operatorlar, acmp.ru saytidagi biror masala shartini ko`rish, ushbu masala uchun yozilgan dasturni sayt orqali tekshirish va tekshirish natijasi to`g`risidagi ma`lumotlar keltirildi.

Bitiruv malakaviy ishining II bobi “Mavzulashtirilgan olimpiada masalalari“ deb nomlanib, unda acmp.ru saytidagi mavzulashtirilgan olimpiada masalalaridan 5 ta mavzudagi masalalar keltirilgan. Bu bob 5 ta bo’limdan iborat bo`lib, har bir bo`limda bir mavzuga doir masalalarning o`zbek tilidagi shartlari va java dasturlash tilidagi dasturlari izohlari bilan berilgan.

Quyidagi mavzudagi masalalar keltirilgan:

1.Oddiy matematika(Простая математика).

2. Uzun sonlar arifmetikasi (Длинная арифметика).

3. Ikki o`lchovli massivlar (Двумерные_массивы).

4. Satrni tahrirlash(Разбор строк).

5. Ketma-ketrliklar va ularni saralash(Сортировка и последовательности).

Ushbu mavzularga doir keltirilgan barcha masalalar acmp.ru saytida tekshirilgan va barcha testlardan o`tgan.

Bitiruv malakaviy ishni bajarishdan asosiy maqsad talabalarga kompyuter tehnologyalaridan foydalangan holda mustaqil shug`ullanishlari uchun o`zbek tilidagi uslubiy qo`llanma yaratishdan iborat. Shu sababli bitiruv malakaviy ishi natijasini uslubiy qo'llanma ko'rinishda tayyorladim. Uslubiy qo'llanmadan talabalar va mustaqil shug`ullanuvchilar internet tizimidagi dasturlash bo`yicha olimpiada masalalarini yechishda kerakli saytlardan foydalanish jarayonlarini, java dasturlash tili yordamida olimpiada masalalarini yechishda tilning imkoniyatlaridan foydalanish usullarini, olimpiada masalalarini yechish algoritmlarini va yechilgan masalalarni olimpiadaga oid saytlarga yuborib to`g`riligini tekshirib ko`rishni o`rganishlari mumkin. Undan tashqari uslubiy qo`llanmadan dasturlashga oid fanlarda ham foydalanish mumkin.

Bitiruv malakaviy ishining kirish qismi 3 betdan iborat, tushuntirish qismi 93 betdan iborat. Bitiruv malakaviy ishida 8 ta adabiyotdan foydalanildi.

Foydalanilgan adabiyotlar ro'yxati.

1. Узбекистон Республикаси "Кадр тайёрлаш миллий дастури" // Баркамол авлод.-Узбекистон тараккиётининг пойдевори.-Т.: "Шарк" нашриёт-матбаа концерни,1997.-Б.31-61
2. Каримов И.А. "Юксак маънавият енгилмас куч". –Т.: маънавият,2008-176 б.
3. Бақоев О, Болтаев Т. Б " JAVA тилида тузилган дастурлар" Бухоро – 2009, 92 б.
4. Бақоев О, Болтаев Т. Б " JAVA тилида ДАСТУРЛАШ" Бухоро – 2009, 64 б.
5. Герберт Шилдт "Полный справочник по Java™ " –М.: Издательский дом " Вильямс"–2007,1024 ст.
6. Дэвид Флэнаган "JAVA СПРАВОЧНИК" четвертое издание Санкт-Петербург-Масква-2004,1039 ст.
7. www\acmp.ru
8. www\dasturchi.uz