

O'ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA
KOMMUNIKASIYALARNI RIVOJLANTIRISH VAZIRLIGI

TOSHKENT AXBOROT TEXNOLOGIYALAR UNIVERSITETI
SAMARQAND FILIALI
KOMPYUTER INJINERING FAKUL'TETI
DASTURIY INJINIRING KAFEDRASI

5330200 –Informatika va axborot texnologiyalari yo'nalishi bo'yicha
bakalavr akademik darajasini olish uchun

BITIRUV MALAKAVIY ISHI

**Mavzu: Dasturlash bo'yicha olimpiadalarni o'tkazishga ko'maklashuvchi
dasturiy vosita yaratish**

Ish kafedraning 2015 yil ____
____dagi ____-sonli
majlisda muhokama qilindi va
himoyaga tavsiya etildi.

Bajardi: 402-guruh talabasi
____ Quvondiqov S.

Kafedra mudiri v. v.b. _____

Ilmiy rahbar:

Kat. O'q. Soqiyev T.

_____ ass. Nuriyev S.

“ ____ ” _____ 2015y.

S A M A R Q A N D – 2015

MUNDARIJA

KIRISH.....	3
1 – bob. DASTURNI TESTLOVCHI AVTOMATLASHTIRILGAN TIZIMNI LOYIHALASH.....	5
1.1 Dasturiy ta’minotni loyihalashning nazariy asoslari	5
1.2. Dasturni testlovchi avtomatlashtirilgan tizimlar haqida.....	14
1.3 Dasturiy ta’minotni loyihalash	16
2-bob. DASTURLASH BO’YICHA OLIMPIADALARNI O’TKAZISHGA KO’MAKLASHUVCHI DASTURIY VOSITA VA UNING TASNIFI	21
2.1. Dasturiy vositani o’rnatish va undan foydalanish qoidalari	21
2.2.Dasturiy vosita tasnifi	25
2.3. Tuzilgan masalani to’g’riligini tekshirish uchun testlar yaratish va testlash	33
2.4. Kompyuter bilan ishlashda hayotiy faoliyat xavfsizligi qoidalari va talablari.	42
XULOSA VA TAKLIFLAR.....	45
ADABIYOTLAR.....	46
ILOVA . SAHIFALAR VA OYNALARNI HOSIL QILISH KODI.....	48

KIRISH

Ishning dolzarbligi. Hisoblash mashinalarining rivojlanishi bilan birgalikda ularning dasturiy ta'minoti ham o'z-o'zidan takomillashib bormoqda. Professional dasturlash qobiliyatiga ega bo'lgan mutaxassislarni tayyorlashda doimiy ravishda dasturlash bo'yicha bilimlarni nazoratdan o'tkazib turish maqsadga muvofiq deb hisoblaymiz. Shuning uchun ham dasturchi yaratgan dasturiy mahsulotining to'g'riligini tekshirish (testlash)da ba'zi bir muammolar tug'iladi. Bu muammolardan biri tuzilgan dasturni tezkorlik bilan testlash, natijaning to'g'riligiga ishonch hosil qilish uchun sarflanadigan vaqt va energiyani tejash muammosi hisoblanadi.

Bu kabi vazifalarni bajarish kadrlar tayyorlash milliy modelini amalga oshirishning bosqichlarida belgilab berilgan bo'lib, uning kelajakdagi istiqboli Prezidentimiz tomonidan ilmiy asoslab berildi. Modelni amaliyotga tatbiq etish o'quv jarayonini texnologiyalashtirish bilan uzviy bog'liqdir. Ilmiy texnikaviy taraqqiyot ishlab chiqarishning ko'p sonli tarmoqlari bilan bir qatorda ta'lim sohasiga ham zamonaviy axborot texnologiyalarini joriy etishni taqozo etmoqda. Shu boisdan, Kadrlar tayyorlash milliy dasturida "o'quv-tarbiyaviy jarayonni ilg'or pedagogik va axborot texnologiyalari bilan ta'minlash" zarurati e'tirof etildi, uning ikkinchi va uchinchi bosqichlarida bajariladigan jiddiy vazifalardan biri sifatida belgilandi". [1]

Ishning maqsadi. Bitiruv malakaviy ishida ihtiyoriy dasturlash tilida tuzilgan dasturni masofadan turib natijasini testlab, yechim to'g'risida yetarli xulosa bera oladigan avtomatlashtirilgan tizimni ishlab chiqish zarur. .

Bu maqsadga erishish uchun quyidagi masalalar yechiladi:

- Ma'lumotlar bazasini loyihalash va ishlab chiqish;
- Dasturni tekshiruvchi qismni ishlab chiqish;
- Masalalar bazasini ishlab chiqish;
- Tizimdan tarmoq texnologiyalari asosida foydalanish imkoniyatini berish uchun interfeys ishlab chiqish;

Ishning yangiligi. Delphi va C++ dasturlash tillari yordamida dasturlash bo'yicha olimpiadalarni o'tkazishga ko'maklashuvchi, ya'ni ishtirokchilar tomonidan tuzilgan dasturni testlovchi dasturiy vosita ishlab chiqildi.

Ishning amaliy ahamiyati. Yaratilgan dasturiy vosita dasturlash bo'yicha fan olimpiadalarida masalalarni to'g'ri yoki noto'g'riligini tekshirib ishtirokchilarning egallagan o'rinlarini aniqlab berishda foydalaniladi.

Shuningdek, O'zbekiston Respublikasi Prezidentining 2012 yil 21 martdagi «Zamonaviy axborot-kommunikasiya texnologiyalarini yanada joriy etish va rivojlantirish chora-tadbirlari to'g'risida» PQ-1730 sonli qarori[2], 2012 yil 1 fevraldagi “Joylarda kompyuterlashtirish va axborot-kommunikasiya texnologiyalarini yanada rivojlantirish uchun shart-sharoitlar yaratish chora-tadbirlari to'g'risida”gi qarorlari Respublikamizda axborot-kommunikasiya texnologiyalari taraqqiyotida yana bir muhim bosqich bo'lib qoladi[3].

Bitiruv malakaviy ishning hajmi va tuzilishi. Ishning matni komp'yuterda yozilgan 65 bet hajmidan iborat bo'lib, uning strukturasi kirish, 2 ta bob, xulosa, foydalanilgan adabiyotlar ro'yxati va 1 ta ilova tashkil qiladi.

1 - bobda Dasturlash bo'yicha olimpiadalarni o'tkazishga ko'maklashuvchi dasturiy ta'monotni loyihalash to'g'risida ma'lumotlar keltirilgan.

2 - bob Dasturlash bo'yicha olimpiadalarni o'tkazishga ko'maklashuvchi dasturiy vosita va uning tasnifiga bag'ishlangan bo'lib, unda dasturiy vositani yaratishda C++, Delphi dasturlarini qo'llash, dasturiy vositaning tuzilmasi, dasturiy vositaning tasnifi, undan foydalanish tartibi va kompyuterda texnika xavfsizligiga rioya qilish qoidalari bayon etilgan.

Xulosa qismida bitiruv malakaviy ishida olingan asosiy natijalar va takliflar keltirilgan.

Adabiyotlar qismida bitiruv malakaviy ishini bajarishda foydalanilgan asosiy adabiyotlar ro'yxati keltirilgan.

Ilovada sahifalar va oynalarni hosil qilish kodi keltirilgan.

1 – bob. DASTURNI TESTLOVCHI AVTOMATLASHTIRILGAN TIZIMNI LOYIHALASH

1.1 Dasturiy ta'minotni loyihalashning nazariy asoslari

Texnikaviy ob'yektni loyihalash ushbu ob'yekt obrazini qabul qilingan forma (shakl)da yaratish, qavta o'zgartirish va tasvirlab berish bilan bog'liq. Ob'yekt yoki uning tarkibiy qismining obrazi inson tasavvurida ijodiy jarayon natijasida yaratilishi yoki inson va EHMlarning o'zaro ta'siri jarayonida ba'zi algoritmlar bo'yichayuzaga kelishi (генерироваться) mumkin.

Loyihalash - loyihalash uchun topshiriq bo'lgan holda bajariladi. Topshiriq jamiyatning qandaydir texnikaviy buyumni olishga bo'lgan ehtiyojini aks ettiradi. Bu topshiriq u yoki liu hujjatlar ko'inishida bo'iadi va ob'yeklning birlamchi bayoni vazifasini bajaradi. Loyihalash natijasini. odatda, ob'yektni berilgan sharoitlarda tayyorlash uchun etarli ma'lumotlarni jamlagan hujjatlarning to'liq komplekti o'taydi. Bu hujjatlar ob'yektning naliyaviy bayoni bo'iadi.

Loyihaviy yeehimlarniig hammasi yoki bir qismi inson va EHMlarning o'zaro ta'siri v o'li biian olinadigari loyihalash aviomatlashtirilgan deb, EHMdan fovdalanilmaydigan loyihalasii esa, avtomatlashtirilmagan loyihalash deb ataladi. Loyihalash - bu ob'yektning birlamchi bayoni va (yoki) uni mavjud qiladigan algoritm asosida berilgan sharoitdr. ham mavjud bo'lmagan ob'yektni yaratish uchun zarur bo'lgan bayonini tuzish ,jarayonidir. Loyihalash berilgan talablarga javob beradigan, yangi buyumni yaratish yoki vangi iarayonn; amalga oshirish uchun zarur va yelarli bo'lgan loyihalanadigan predmet bayonini oiish maqsadidagi izlartish, tadqiqot, hisob va konstruktsiyalash bo'yicha ishlar majmuini o'z icliiga oladi. Loyihalash - bu chuqur ilmiy bilimlarga va ijodiy izlanishlarga hamda ma'lum sohada to'plangan tajriba va ko'nikmalardan foydalanishga asoslangan, lekin sermashaqqat oddiy ishlarni bajarish zarurati bo'lgan inson biinyodkorlik faolivatining murakkab, o'ziga xos turidir.

Loyihalashni avtomatlashtirish deganda loyihani ishlab chiqish jarayonini bajarishning shunday usuli lushuniladiki, bunda loyihalash protseduralari va operatsiyalari loyihalovchining EHM bilan chambarchas muloqotida amalga

oshadi. Loyihalashni avtomatlashtirish hisoblash texnikasi vositalaridan muntazam ravishda foydalanishni nazarda tutadi; bunda loyihalovchi va EHM orasidagi funktsiyalarni ratsional taqsimlash va masalalarni mashinada yechish metodlarini asosli tanlash lozim.

ALTni yaratish uchun:

- matematik metodlar hamda hisoblash texnikasi vositalarini qo'llash asosida loyihalashni takomillashtirish;
- izlash, ishlov berish va informatsiya (ma'lumot)ni chiqarish jarayonlarini avtomatlashtirish;
- optimallashtirish va ko'p variantli loyihalash metodlaridan foydalanish;
- loyihalalanayotgan ob'ektlar, buyumlar va materiallarning matematik modellarini samarali qo'llash;
- ob'ektlarni avtomatlashtirilgan loyihalash uchun zarur bo'lgan, ma'lumotnoma tavsifidagi tizimlashtirilgan ma'lumotlarga ega ma'lumotlar bankini yaratish;
- loyiha hujjatlarini shakllantirish (rasmiylashtirish) sifatini oshirish;
- ijodiy bo'lmagan ishlarni avtomatlashtirish hisobiga loyihalovchilar mehnatining ijodiy ulushini oshirish;
- loyihalash metodlarini unifikatsiyalash va standartlashtirish;
- ALT sohaqidagi mutaxassislarni tayyorlash va qayta tayyorlash;
- loyihalovchi bo'limlarning turli darajadagi hamda vazifasi har xil bo'lgan avtomatlashtirilgan tizimlar bilan mustahkam aloqada ishlashi zarur.

Avtomatlashtirilgan loyihalash tizimi (ALT) - avtomatlashtirilgan loyihalashni bajaruvchi loyihalovchi tashkilot yoki mutaxassislar jamoasi bilan bog'langan avtomatlashtirilgan loyihalash vositalarining majmuidir. ALT texnikaviy vositalar hamda matematik va dasturaviy ta'minlashni birlashtiradi; matematik va dasturaviy ta'minot muhandislik loyihalash va konstruksiyalash masalalarining xususiyatlarini maksimal hisobga olgan holda tanlanadi. ALTda muhandisning EHM bilan operativ bog'lanishi vositalari, maxsus muammoli-

yo'naltirilgan tillar va informatsion- ma'lumot bazasi qo'llanilishi hisobiga dasturlardan foydalanish qulayligi la'minlanadi.

ALTning asosiy vazifasi - ob'yekt va uning tarkibiy qismlarini loyihalashni avtomatlashtirilgan tarzda bajarishdir. ALT va uning tarkibiy qismlarini yaratishda tizimiy birlik, bir- biriga mos kelish, tipik xususiyatlarga qarab tip va turlarga bo'lish hamda rivojlanish printsiplariga amal qilish lozim.

Tizimiy birlik printsipi. Loyihalanayotgan ob'yektning alohida elementlari va ob'yektni to'liq loyihalashda tizimning bir butunligini va tizimiy «yangilik»ni ta'minlavdi.

Bir-biriga mos kelish printsipi ALTning tarkibiy qismlarining birgalikda ishlashini ta'minlaydi va ochiq tizimni bir butunlikda saqlaydi.

Tipik xususiyatlarga qarab tip va turlarga bo'lish printsipi ALTning tipiklashgan va unifikatsiyalashgan elementlarini yaratish va ulardan foydalanishga e'tiborini qaratadi.

Rivojlanish printsipi ALT asosiy qismlarining to'ldirib borilishini, takomillashtirilishini va yangilanib borishini hamda darajasi va funktsional vazifasi turlicha bo'lgan avtomatlashtirilgan tizimlar bilan birgalikda ishlashini ta'minlaydi.

ALT - inson-mashina tizimi. EHM yordamida hamma tuzilgan va loyilayotgan loyihalash tizimlari avtomatlashtirilgan tizimlarga kiradi. Ularda loyihani texnikaviy vositalar yordamida ishlayotgan inson muhandis salohiyatli o'rin egallaydi. ALTda inson birinchidan formalizatsiya qilinmagan masalani va ikkinchidan insonning evrestik qobiliyatlari asosida samaraliroq echiladigan masalalarni vechadi.

ALT - ierarxik tizim. U hamma darajalarda loyihalashni avtomatlashtirishga kompleks yondoshuvni amalga oshiradi. ALT qo'llanilganida loyihalashga blokli-ierarxik yondoshuv saqlanib qolishi kerak. Loyihalashning ierarxik darajalari ierarxik nimitizim ko'rinishida ALTning maxsus dasturaviy ta'minoti (DT) strukturasida o'z aksini topadi.

Loyihalash - tadqiqot, hisoblash va konsruktorlik tavsifidagi ishlar kompleksim bajarish asosida ob'yektning birlamchi bayonini natijaviy bayonga o'zgartiradigan jarayondir.

Birlamchi bayonni natijaviy o'zgartirish oraliq bayonlarni tug'diradi; ular loyihalash tugaganini aniqlash yoki uni davom ettirish yo'llarini tanlash maqsadida qarab chiqiladigan predmet vazifasini o'taydi. Bunday bayonlarni loyihali yechimlar deyishadi.

ALT - ochiq va rivojlanuvchi tizimdir. ALT vaqt o'tishi bilan o'zgaruvchi tizim bo'lishi kerakligiga kamida ikkita sabab bor. Birinchidan, ALT kabi murakka'o ob'ektni ishlash uzoq muddatni egallaydi, shuning uchun ALT tizimining qismlari tayyor bo'la borgan sari ularni ekspluatatsiyaga kiritish iqtisodiy nuqtai nazardan foydalidir. Ekspluatatsiyaga kiritilgan tizimning bazaviy varianti keyinchalik kengaytirib boriladi. Ikkinchidan, hisoblash texnikasi va hisoblash matematikasining doimiy progressi yangi, ancha takomillashgan matematik modellar va dasturlar paydo bo'lishiga olib keladi; ular eskirgan, samaradorligi kam bo'lgan analoglarni almashtirishi kerak. Shu sababli ALT ochiq tizim bo'lishi, ya'ni yangi metod va vositalarni ulash qulay bo'lgan qobiliyatga ega bo'lishi zarur.

ALT - unifikatsiyalashgan modullardan maksimal foydalaniladigan ixtisoslashtirilgan tizimdir. Yuqori effektivlik va universallik talablari, o'zaro, bir-biriga qarama-qarshidir. Loyihalash vazifalarini yechishda kam vaqt va materiallar sarf bo'lishida ifodalanadigan ALTning yuqori samaradorligiga tizimlarning ixtisoslashtirilishi hisobiga erishiladi. Ixtisoslashtirilgan ALTni ishlab chiqishga ketadigan sarfni kamaytirish uchun ularni unifikatsiyalashgan tarkibiy qismlardan maksimal foydalangan holda tuzish maqsadga muvofiqdir. Unifikatsiyalashning zaruriy sharti - turli texnikaviy ob'ektlarni modellashda, tahlil va sintez qilishda umumiy holatlarni qidirishdadir.

ALT tarkibi va strukturasi. Nimtizimlar ALTning tarkibiy strukturaviy qismi bo'lib, loyihalovchi tashkilotning tashkiliy strukturasi bilan chambarchas

bog'lanadi; ularda ixtisoslashgan vositalar kompleksi yordamida ALTning funksional tugal masalalar ketma-ketligi yechiladi.

Vazifasi bo'yicha nimtizimlarni loyihalovchi va xizmat ko'rsatuvchilarga ajratishadi.

Loyihalovchi nimtizimlar. Ular ob'yektga yo'nalgan bo'ladi va loyihalashning ma'lum bosqichini yoki o'zaro bevosita bog'langan loyihalash masalalarining bir guruhini amalga oshiradi.

Loyihalovchi nimtizimlarga misollar: buyumlarni eskiz loyihalash, korpus detallarini loyihalash, mexanik ishlov berish texnologik jarayonlarini loyihalash.

Xizmat ko'rsatuvchi nimtizimlar. Bunday nimtizimlar umumiy tizimga ishlatiladi va loyihalovchi nimtizimlar o'z funktsiyalarini bajarishda ulanib qo'llab-quvvatlashni hamda ularda olingan natijalarni shakllantirish, uzatish va chiqarishni ta'minlaydi.

Xizmat ko'rsatuvchi nimtizimlarga misollar: avtomatlashtirilgan ma'lumotlar banki, hujjatlashtirish nimtizimlari, grafik kiritish-chiqarish nimtizimi.

ALTning tizimiy birligi bir-biri bilan o'zaro bog'langan modullarning mavjudligi hamda o'zaro bog'lanishni amalga oshiruvchi interfeyslar tizimi kompleksi bilan ta'minlanadi; modullar loyihalanadigan ob'yektni butunligicha belgilaydi. Loyihalovchi nimtizimlar ichidagi tizimiy birlik ushbu nimtizimda loyihaviy yechimi olinishi kerak bo'lgan ob'yekt qismining yagona informatsion modeli mavjudligi bilan ta'minlanadi.

Amaliy masalalarda loyihalanadigan ob'yekt modellarini shakllantirish va ulardan foydalanish avtomatlashtirilgan loyihalash tizim (yoki nimtizim)lari vositalari kompleksi (ALTVK) bilan amalga oshiriladi.

ALTVK. tizimining strukturaviy qismlari bo'lib turli vositalar komplekslari hamda tashkiliy ta'minlash komponentlari xizmat qiladi. Vositalar kompleksi - ALTning mos loyihalovchi va (yoki) xizmat ko'rsatuvchi nimtizimlaridan foydalaniladigan, tirajlash uchun mo'ljallangan va ma'lum klass (tur, rukum) ob'ektlarini loyihalashga yo'nalgan va (yoki) unifikatsiyalashgan protseduralarni bajarishga mo'ljallangan komponentlar va (yoki) vositalar kompleksi majmuidir.

Vositalar kompleksi tayyorlanadigan, tirajlanadigan va ALT tarkibida qo'llaniladigan sanoat buyumlariga kiradi va spetsifikatsiyalanadigan buyumlar kabi hujjatlantiriladi.

ALT vositalari kompleksi va komponentlarining turlari (1.1-rasm). Vositalar kompleksini ikki turga: bir turdagi ta'minlash vositalari kompleksiga (texnikaviy, dasturaviy, informatsion) va kombinatsiyalashgan vosilalar kompleksiga ajratishadi.

Bir turdagi ta'minot vositalari komplekslari bir turdagi ta'minlash komplekslaridan va (yoki) komponentlaridan tarkib topadi; kombinatsiyalashgan vositalar komplekslari esa - lix turdagi ta'minlash komplekslari va komponentlari majmuidan tashkil bo'ladi. Vazifasi ishlab-chiqarish- texnikaviy bo'lgan mahsulotlarga taalluqli kombinatsiyalashgan ALT VKlar ikki turga bo'linadi:

- dasturaviy-metodik kompleks (DMK);
- dasturaviy-texnikaviy kompleks (DTK).

Dasturaviy metodik kompleks loyihalash ob'yekti (ob'yektning bir yoki bir necha qismi yoki bir butun ob'yekt) bo'yicha tugal loyiha yechimini olish yoki unifikatsiyalashgan protseduralarni bajarish uchun zarur bo'lgan dasturaviy, informatsion va metodik ta'minotlar (matematik va lingvistik ta'minotlar komponentlari bilan birga) komponentlarining o'zaro bog'langan majmuidan iborat.

Vazifasi bo'yicha DMKlar umumtizimiy DMKlarga va bazaviy DMKlar o'z navbatida muammoga yo'nalgan va ob'yektga yo'nalgan DMKlarga bo'linadi.

Dasturaviy-texnikaviy kompleks DMKlarning texnikaviy ta'minotning komplekslari va (yoki) komponentlari bilan o'zaro bog'langan majmuidan iborat.

Vazifasi bo'yicha DTKlar avtomatlashtirilgan ish joyi (AIJ) va markaziy hisoblash komplekslari (MHK)ga bo'linadi.

Vositalar komplekslari o'zlarining hisoblash va informatsion resurslarini birlashtirib nimitzim yoki butun tizimlarning lokal hisoblash tarmoqlarini tashkil qilishi mumkin.

Dasturaviy informatsion, metodik, matematik, lingvistik va texnikaviy ta'minot tuiiarining komponentlari vositalar komplekslarining tarkibiy qismi hisoblanadi.

ALTVK funktsiyalarini samarali bajarishi vositalar komplekslari tarkibiga kiruvchi komponentlarni sotib olinadiganlari bilan o'zaro moslashuvini ta'minlagan holda ishlab chiqish hisobiga erishilishi kerak.

Umumtizimiy DMKlar dasturaviy, informatsion, metodik va boshqa turdagi ta'minotlarni o'z ichiga oladi. Ular boshqaruv, nazorat, hisoblash jarayonini rejalashtirish, ALT resurslarini taqsimlashni bajarish va nimtizim yoki butun ALT uchun umumiy bo'lgan boshqa funktsiyalarni amalga oshirish uchun mo'ljallangan. Umumtizimiy DMKlarga misollar: monitor tizimlari, ma'lumotlar bazalarini (MB) boshqarish tizimlari, informatsion-qidiruv tizimlari, mashina grafikasi vositalari, dialogli rejimni ta'minlovchi nimtizimlar va h.k.

ALTda texnikaviy vositalar funktsiyalarini bajarishin: boshqarivchi monitor tizimlari. Monitor tizimlarining asosiy funktsiyalari:

- talab qilinadigan va mavjud resurslar masalalari paketini nazorat qiluvchi topshiriqlarni ustuvorligi va navbat nomeri o'rnatilgan holda ma'lumotlar bazasiga kirish huquqini shakllantirish;
- topshiriqlar va masalalarni boshqarish tillarining yo'riqlariga ishlov berish hamda uzilishlarga boshqarishni ilib olib, uzilish sababini tahlil qilib va uni loyihalovchiga tushunarli terminlarda izohlab reaksiya (sezib ta'sir) qilish;
- nimtizimlar parallel ishlagan sharoitlarda dialogli va interaktiv-grafik hamrohligini tashkil qiluvchi masalalar oqimiga xizmat ko'rsatish;
- avtomatik rejimlarda loyihalash operatsiyalarining bajarilishi sifatini tanlil qilgan, bosqichning yoki marshrutning davom etishi mezonlari tekshirilgan, marshrutning alternativ qaytarilishi variantlarini tanlagan holda loyihalashni boshhaarish;
- tizimni ekspluatatsiya qilish statistikasini olib borish va optimallashtirish;
- topshiriqlar masalalar va nimtizimlar, rejali topshiriqlar va joriy ko'rsatmalar va so'rovlar ustuvorligini hisobga olgan holda ALT resurslarini taqsimlash;

- resurslar va ma'lumotlarni ruxsat etilmagan kirishdan va nazarda tutilmagan ta'sirlardan himova qilish.

ALTda informatsion-qidiruv tizimlar (IQT) quyidagi funktsiyalarni bajaradi:

- informatsion fond (информотека)т ma'lumotlar bilan to'ldirish;
- raqamli ma'lumotlarga arifmetik ishlov berish va matnlarga leksikaviy ishlov berish;
- informatsion so'rovlarga zarur bo'lgan ma'lumotlarni qidirish maqsadida ishlov berish;
- chiquvchi ma'lumotlarga ishlov berish va chiquvchi hujjatlarni shakllantirish.

IQTning xususiyati shundaki, ularga kelgan so'rovlar dasturaviy yo'l bilan emas, balki bevosita foydalanuvchi tomonidan shakllantiriladi va monitorga tushunarli bo'lgan formallashgan tilda emas, balki tayanch so'zlar ketma-ketligi ko'rinishida «deskriptor»lar deb nomlanuvchi tabiiy tilda shakllanadi. Saqlash uchun qabul qilingan hamma bayonlarda bo'lgan deskriptorlar ro'yxati deskriptorlar lug'atini tashkil qiladi va qidiruvchi yo'riqlarni shakllantirishga mo'ljallangan.

Deskriptorliga nisbatan ancha murakkab bo'lgan IQTlar ham mavjud. Ularda informatsion- qidiruv tili katta ahamiyatga ega; bu tilda informatsion ob'yektlar orasidagi semantik munosabatlar hisobga olinadi. Bu esa noto'g'ri tanlanadigan til qurilmalari sonini kamaytirish imkonini beradi, so'rovlarga ishlov berish esa ma'no mos kelishi mezonlari asosida bajariladi.

Ma'lumotlar banki katta ALTlarda informatsiyani tashkil qilishning yuqori shakli hisoblanadi. Ular - muammoli-yo'nalgan informatsion-ma'lumotnomalar tizimlaridir. Bu tizimlar kiritishning muayyan vazifalariga bog'liq bo'lmagan zarur informatsiyalarni kiritishni, informatsion massivlar saqlanishini hamda foydalanuvchilar yoki dasturlar so'rovi bo'yicha zarur bo'lgan informatsiyani chiqarishni ta'minlaydi. Ma'lumotlar bankida faktografik ko'rinishdagi informatsiyadan foydalaniladi.

Ma'lumotlar bazasini boshqarish tizimi (MBBT) - ma'lumotlar strukturasi ko'rinishida tashkil qilingan informatsion baza bilan ishlashni ta'minlaydigan dasturaviy-metodik kompleksdir.

MBBT quyidagi asosiy funktsiyalarni bajaradi: ma'lumotlar bazasini aniqlab olish, ya'ni sxemalarning kontseptual tashqi va ichki darajalarini bayon qilish; ma'lumotlarni bazagayozish; ma'lumotlarga o'zgartish va qo'shimchalar kiritish, ularni qayta tashkil qilishni bajargan holda ularning saqlanishini tashkil qilish; ma'lumotlarga kira olishni ta'minlash (qidirish va chiqarib olish).

Ma'lumotlarni tanib olish va ularga kirish uchun MBBTda til vositalari mavjud. Masalan, ma'lumotlar strukturasi bayonidan tashkil topgan ma'lumotlarni tanib olish tili yordamida ta'minlanadi. Ma'lumotlarga kirish funktsiyasi ma'lumotlarni manipulyatsiya qilish tili va so'rov qilish tili yordamida amalga oshadi. Qo'llab (tutib) turiladigan strukturalar bo'yicha MBBTning ierarxik, tarmoqli va nisbiy (relyativ) turlari bo'ladi.

Mashina graftkasining dasturaviy-metodik komplekslari (DMK) foydalanuvchining EHM bilan muloqotida grafik informatsiya almashinuvini, geometrik masalalarni yechishni, tasvirlarni shakllantirishni va grafik informatsiyani avtomatik ravishda tayyorlashni ta'minlaydi. Foydalanuvchining EHM bilan grafik muloqoti («kirishning grafik metodi») kirish-chiqish nimdasturlariga asoslanadi; bu nimdasturlar kiritish-chiqarish qurilmalaridan olinadigan komandalarning qabul qilinishini va ularga ishlov berilishini hamda ushbu qurilmalarga boshqaruv ta'sirlarining chiqarilishini ta'minlaydi. Geometrik masalalar yechimi (grafik modellash) grafik informatsiyani qayta o'zgartirishga keltiriladi; bu o'zgartirish qurish, burish, masshtablash va sh.k. turlardagi elementar grafik operatsiyalarni u yoki bu ketma-ketlikda bajarilishida ifodalanadi. Grafik modellash uchun DMKIardan foydalaniladi: ularda alohida elementar grafik operatsiyalardan tashqari uch o'lchamli tasvirlarni grafik qayta o'zgartirishlar, proektsiya, kesim va h.k.larni qurish protseduralari amalga oshirilishi mumkin. Grafik o'zgartirishlar DMKsida ba'zi tez-tez foydalaniladigan tasvirlarni

shakllantirish, grafik ma'lumotlar bazasini boshqarish, grafikaviy nimdasturlarni sozlash uchun vositalar odatda nazarda tutiladi.

1.2. Dasturni testlovchi avtomatlashtirilgan tizimlar haqida

Hozirgi kunga qadar dastur natijasini testlovchi o'nlab avtomatlashtirilgan tizimlar ishlab chiqilgan. Bular jumlasiga, Z-traing, Campion, Caribbean Online Judge, Fuzhou University Online Judge, Harbin University Online Judge, Infoarena, Lviv National University Online Judge, Moscow Problem Archive with Online Judge, Peking University Online Judge, Saratov State University Problem Archive with Online Judge, Sphere Online Judge, Tianjin University Online Judge, Timus Online Judge va boshqalarni keltirish mumkin. Bu tizimlar o'zaro ularning ma'lumotlar bazasidagi masalalar arxivi soni, qo'llash mumkin bo'lgan dasturlash tillari soni kabi jihatlari bilan farqlanadi. Bunday tizimlarning afzalligi, qisqa vaqt muddatida berilgan yechimni testdan o'tkazib, uning to'g'ri yoki xatoliklar haqida axborotni dasturchiga taqdim etishdan iboratdir. Bundan tashqari dasturning ishlash vaqti va xotira hajmini juda qisqa vaqtda yuqori aniqlikda tekshirib berishi ham mumkin[4][5].

Yana shuni alohida ta'kidlash mumkinki, yuqoridagilar bilan birgalikda o'qituvchi yoki murabbiy vazifasini to'liq bajara olmasligi kabi kamchiliklarni ko'rsatish mumkin. Lekin bu tizimlarni o'quv jarayonida qo'llash uchun yaratiladigan variantida o'qituvchi to'liq administrator sifatida tizimga murojaat qilishi va natijalarni bir necha usullarda tahlil qilish imkoniyatiga ega bo'ladi. Dastur natijasini testlovchi avtomatlashtirilgan tizimni ishlab chiqish quyidagi bosqichlarda amalga oshiriladi[6][7].

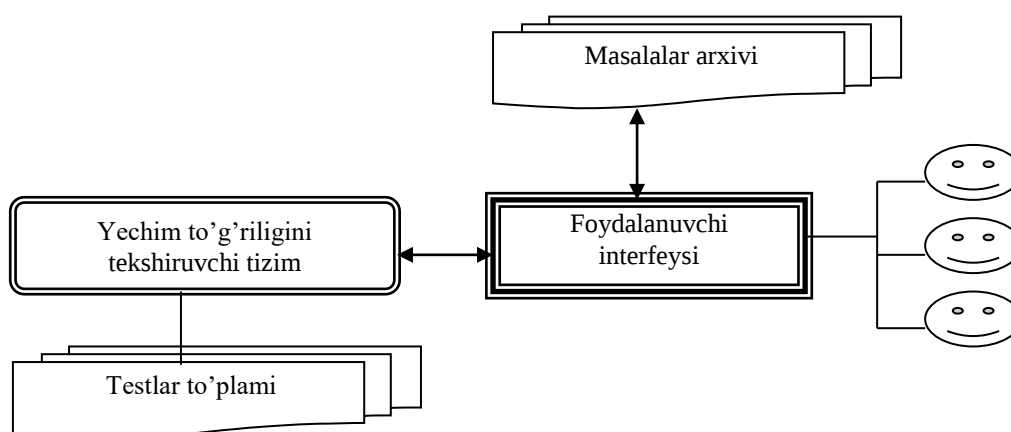
1-bosqich. Ma'lumotlar bazasini loyihalash va ishlab chiqish. Bu bosqichda tizimda qayta ishlanadigan barcha ma'lumotlar maxsus ma'lumotlar bazasida saqlanishi e'tiborga olinishi talab qilinadi. Jumladan, dastur tuzish uchun taklif etiladigan masalalar, uning yechimlari va unga tegishli bo'lgan dastur kodi, dasturning ishlash vaqti, xotira hajmi arxivlangan holda ma'lumotlar bazasida saqlanishi zarur. Tizim ish holatida unga kiritilgan yechimning tekshirilishi

natijalari to'g'risidagi barcha ma'lumotlar ham tizimning bazasiga saqlab qo'yilishi va tahlil uchun administratorga taqdim etilishi lozim bo'ladi.

2-bosqich. Tekshiruvchi qismni ishlab chiqish. Bu bosqichda ma'lumotlar bazasida mavjud masalalar uchun taklif etiladigan yechimning dastur kodi to'g'ri yozilganligi, uning ishlash vaqti va xotira hajmini hisoblab beruvchi tizimni yaratish ko'zda tutiladi. Masalalar yechimini foydalanuvchi tomonidan bir nechta dasturlash tillarida taklif etilishi e'tiborga olinib, tekshiruvchi tizimning ushbu dasturlash tillariga mos variantlari ishlab chiqiladi.

3-bosqich. Masalalar bazasini ishlab chiqish. Bu bosqichda dasturlashga oid fanlarda amaliy va laboratoriya mashg'ulotlari uchun tavsiya etiladigan dasturlashga oid masalalar bazasi va ularga mos masala yechimining to'g'riligini testdan o'tkazish uchun test variantlari bazalarini ishlab chiqish ko'zda tutiladi. Shu bilan birgalikda har bir masalaning taklif etiladigan dasturlash tillaridagi to'g'ri yechim kodlari ham bazaga alohida fayl sifatida kiritib qo'yilishi zarur.

4-bosqich. Tizimdan tarmoq texnologiyalari asosida foydalanish imkoniyatini berish uchun Web-interfeysni ishlab chiqish. Bu bosqichda server-kliyent texnologiyasi asosida ishlovchi va yuqoridagi bosqichlarda ishlab chiqilgan ma'lumotlar bazasini boshqaruvchi maxsus Web-interfeys ishlab chiqish ko'zda tutilgan.



Dastur bajarilishini tekshiruvchi avtomatlashtirilgan tizimning umumiy sxemasi

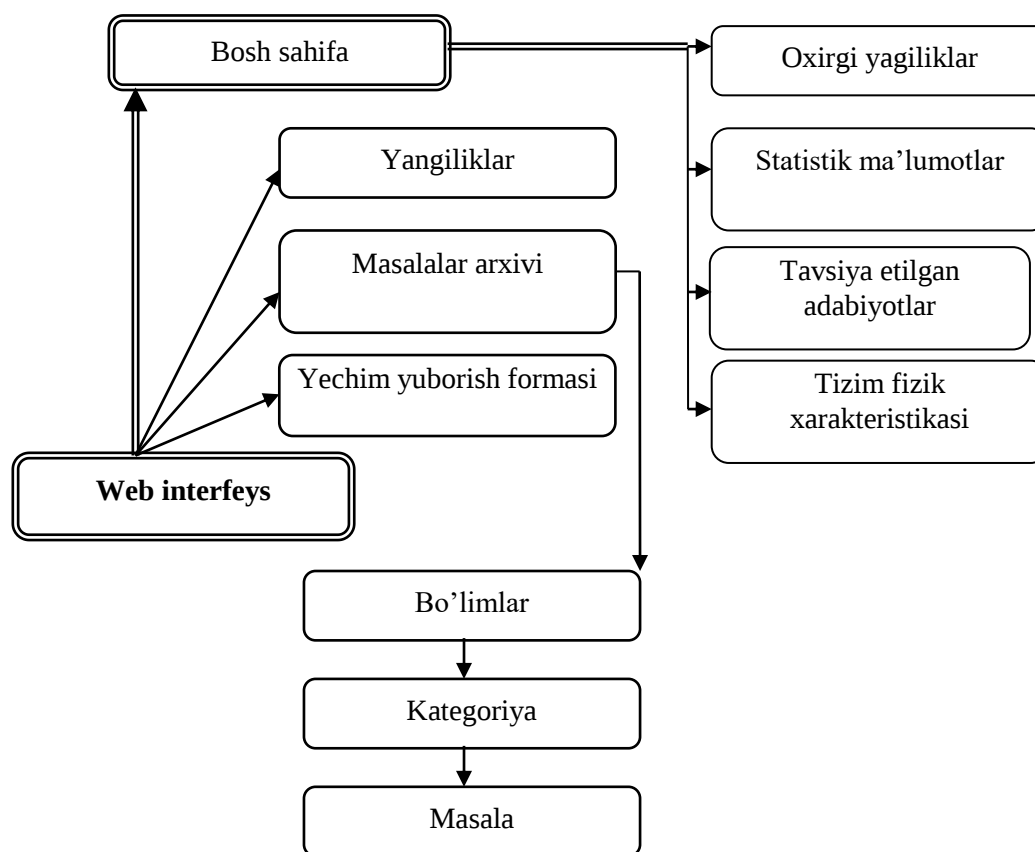
Bu tizimlarning asosiy elementi dastur natijasining to'g'riligini tekshiruvchi blok hisoblanadi. Bu qism tizimda foydalanuvchiga taklif etilgan masalaning yechimlari va uni testlar bazasi bilan solishtiruvchi testlovchi dasturlari bazasi mavjud[8][9][10].

1.3 Dasturiy ta'minotni loyihalash

Dasturlash bo'yicha masala yechimini tekshiruvchi avtomatlashtirilgan tizimni ishlab chiqish quyidagi maqsadlarni o'z ichiga oladi:

- 1) Dasturlash fani bo'yicha bakalavr va magistrning o'zlashtirish samaradorligi sifatini monitoring qilish.
- 2) dasturlash bo'yicha olimpiadalarga tayyorlash
- 3) dasturlash bo'yicha olimpiadalarni o'tkazish
- 4) dasturchilar malakasini oshirish.

Dastur bajarilishini testlovchi tizimning funksional sxemasi tasvirlangan.



Tizim Web-interfeysi.

Ishlab chiqilgan tizim ikkita rejimda ishlay oladi:

1) Ta'lim muassasasi lokal tarmog'ida fanlarni o'qitishda amaliy va laboratoriya mashg'ulotlari jarayonida foydalanish rejimi;

2) Internet tarmog'ida on-layn rejimida bilimlarni mustaqil tekshirish uchun qo'llaniladigan rejimi.

Tizim kliyent – server texnologiyasi asosida (taqsimlanuvchi rejimda) ishlaydi. Tizimni ishlab chiqishda Web-texnologiyalar asoslangan. Shuning uchun tizim ishlashi uchun minimal talablar quyidagicha bo'lishi zarur:

1. Windows XP/7 operasion tizimi;
4. Delphi (Paskal), Java, C, C, C++, C#, kompilyatorlari;
5. Lokal tarmoq va uning vositalari.
6. Pentium IV va undan yuqori;
7. Tezkor xotira 512 Mb va undan yuqori;
8. Qattiq diskdan 750 Mb va undan yuqori bo'sh joy.

Tizimdan foydalanuvchilar ishlash algoritmi: *talaba rejimida ishlash algoritmi*:

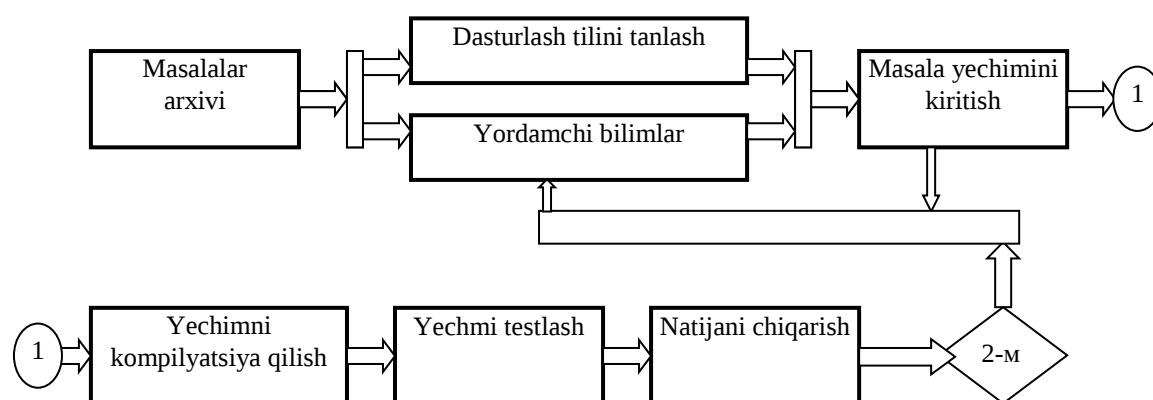
1. Avtorizasiya qilish;
2. Topshiriqlarni olish;
3. Yechimlarni uzatish (yechimlar MB da saqlanadi);
4. Chekker (testlovchi modul) yechimni tekshirish uchun MBdan oladi;
5. Chekker olingan yechim haqidagi axborot asosida uni semantik tahlil qiladi;
6. Chekker yechimni kompilyatorga uzatadi;
7. Agar yechimning kompilyasiyasida xatolik ro'y bersa, xatolik kodini saqlab qolib, chekker o'z ishini to'xtatadi;
8. Kompilyasiya qilingan yechim testdan o'tkaziladi;
9. Chekker birinchi testni ushbu yechim uchun qabul qiladi va h.k.;
10. Test topshiriqlarini faylda saqlab qo'yadi (standart kirish, 0-nomerdagi potok);

11. Chiquvchi ma'lumotlar uchun fayl hosil qiladi (standart chiqish, 1-nomerdagi potok);
12. Qismjarayon hosil qilinadi va yechim uchun ishga tushiriladi;
13. Bu qismjarayon xotira hajmi va prosessor vaqtini nazorat qiladi va cheklangan limitdan oshish holati bo'lganda jarayon to'xtatiladi;
14. Yechim bajarilgandan keyin, natijalar MBdagi etalon bilan taqqoslanadi;
15. Taqqoslashlar MB dagi barcha testlar uchun o'rinli bo'lsa, keyingi patokdagi yechimni tekshirishga o'tadi;
16. Yechim barcha testlardan o'tkazilganda to'g'ri ekanligi tasdiqlangan holda foydalanuvchi – talabaga bu haqida ma'lumot chiqariladi.

O'qituvchi rejimida ishlash algoritmi:

1. Avtorizasiya qilish;
2. Faoliyat turini tanlash (statistik ma'lumotlarni ko'rish, masalalarni tahrirlash, masalalar uchun test topshiriqlarini tahrirlash, yordamchi ma'lumotlarni tahrirlash, talabalar guruhini boshqarish);
3. Agar statistika tanlangan bo'lsa, o'qituvchi har bir talabaning individual natijalarini tahlil qilishi, ularning o'rganishi zarur bo'lgan bilimlarni tahrirlashi, natijalarni taqqoslab, guruhlar yoki kurs bo'yicha talabaning o'zlashtirish darajasini baholashi mumkin;
4. Agar masalalarni tahrirlash tanlansa, o'qituvchi yangi masala kiritishi yoki bazada mavjud masalalarning shartlarini o'zgartirishi mumkin;
5. Agar test topshiriqlarini tahrirlash tanlansa, u holda o'qituvchi testlar sonini ko'rish, zarur holda uni ko'paytirish yoki kamaytirishi, masalalarning murakkablik darajasiga qarab yangi testlarni qo'shishi mumkin;
6. Yangi testlar qo'shilgandan keyin, ularning tekshirib ko'rish va natija to'g'ri bo'lganda uni testlar to'plamiga qo'shib qo'yishi mumkin;
9. Agar yordamchi ma'lumotlarni tahrirlash tanlansa, o'qituvchi bu ma'lumotlarni tahrirlashi yoki yangi maqolalar qo'shish imkoniyatiga ega bo'ladi;
10. Agar talabalar guruhini boshqarish tanlansa, o'qituvchi tizimdagi talabalar guruhiga yangisini kiritishi yoki olib tashlashi mumkin[11].

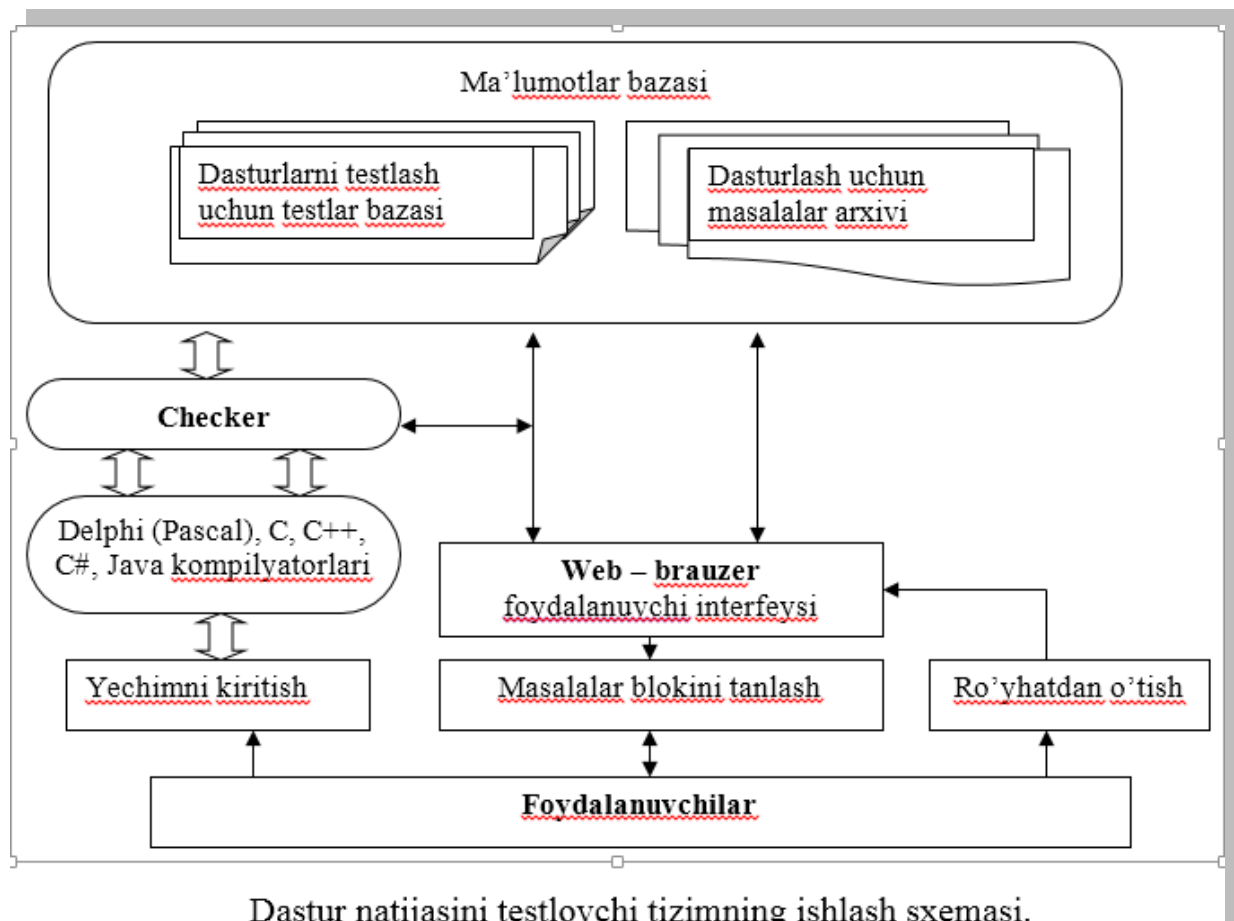
Ushbu algoritmlar asosida ishlovchi tizimga foydalanuvchi kirganda berilgan ro'yxatdan dasturlash tilini tanlab oladi. Mavzular tanlangandan keyin foydalanuvchiga topshiriqni tanlash taklif etiladi va shundan keyin topshiriq shartlari taqdim etiladi. Foydalanuvchi tanlab olingan topshiriqni yechimini tekshirish uchun tizimga uzatishi mumkin. Tizim esa, o'z navbatida yechimni tahlil qilish va maxsus testlardan o'tkazishga kirishadi hamda foydalanuvchiga natijalarni e'lon qiladi[12].



Bunda 2-m.– foydalanuvchining 2-modul bo'yicha ma'lumotlarga o'tishi yoki oldingi modul bo'yicha mustaqil o'rganishga qaytishini bildiruvchi shart

Tizimning ishlash strukturasi.

Dastur natijasini testlovchi avtomatlashtirilgan tizimning ishlash prinsiplari quyidagicha: har bir tizimga kirish ruxsatnomasiga ega bo'lgan foydalanuvchi (talaba, o'qituvchi yoki administrator) uchun maxsus interfeys mavjud. Foydalanuvchilarning tizim bilan ishlashini faqat ma'lumotlar bazasidagi mavjud parametrlar orqali cheklashi mumkin. Foydalanuvchi interfeysi ma'lumotlar bazasi bilan bog'liq holda foydalanuvchining hisob yozuvlari, masalalar arxivi, yechimlarni tekshirish uchun testlar to'plami va boshqa tizimli ma'lumotlar bilan ishlaydi. Masalalarni yechimlarini tekshiruvchi Chekker alohida tizim administratori tomonidan ishga tushiriladi, bu chekker yechimni qabul qilib tekshiruv amallarini bajaradi[13].



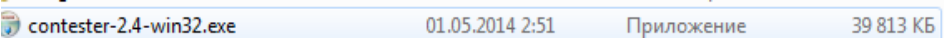
Tasvirlangan dastur natijasini testlovchi tizimning asosini chekker qismtizimi tashkil etadi. Bu qismtizim quyidagilardan tashkil topgan:

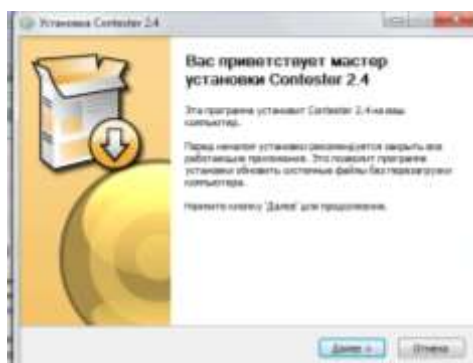
- 1) Aynan tanlangan masala yechimini kompilyatorga uzatish moduli;
- 2) Kompilyasiya qilingan yechimni ma'lumotlar bazasiga joylashtiruvchi modul;
- 3) Kompilyasiya qilingan yangi yechimlar potogini navbatga qo'yish;
- 4) Ma'lumotlar bazasidagi testlar bazasi bilan yechim (kompilyasiya qilingan)ni taqqoslash va natijani foydalanuvchiga uzatish moduli.

Yuqorida sanab o'tilgan asosiy modullarning tizimning ishlash vaqtidagi bajarilish algoritmlari tahlil qilamiz. Bu algoritmlar dasturlash tilida quyidagicha ko'rinishlarga ega, ya'ni ushbu qismtizim Delphi dasturlash muhitida ishlab chiqilgan.

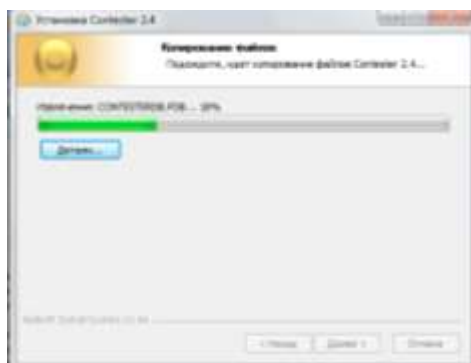
2-bob. DASTURLASH BO'YICHA OLIMPIADALARNI O'TKAZISHGA KO'MAKLASHUVCHI DASTURIY VOSITA VA UNING TASNIFI

2.1. Dasturiy vositani o'rnatish va undan foydalanish qoidalari

Dasturiy vosita install faylini ishga tushirish natijasida o'rn'tilish jarayoni boshlanadi:  va quyidagi oyna hosil bo'ladi:



Dalle tugmasini bosish natijasida dasturiy vositani o'rnatish boshlanadi:



Dasturiy vosita o'rnatilgandan keyin uni  tugmasini bosish natijasida ishga tushiriladi. Natijada ekranning o'ng past qismida quyidagi oyna hosil bo'ladi:



Ushbu oynadan *start* tugmasi dasturiy vositani ishga tushiradi va dasturiy vositadan brauzer orqali foydalaniladi. Bundan tashqali portni to'g'rilash mashina sozlamalarini ko'zdan kechirish imkoniyatlari mavjud. Agar foydalanuvchi o'z kompyuteri orqali dasturiy vositaga kirmoqchi bo'lsa ushbu oynaning *brauzerni ochish* tugmasi yordamida kiriladi. Agar foydalanuvchi dasturiy vositaga tarmoq orqali kirmoqchi bo'lsa unda dasturiy vosita o'rnatilgan kompyuterning IPAdressi orqali murojaat qilinadi.

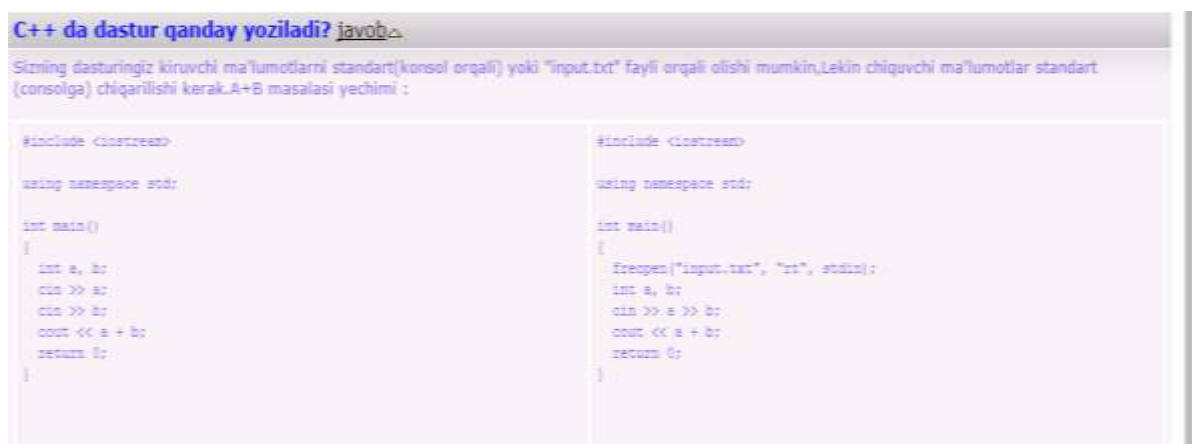


Yuqoridagi dasturiy vositaning asosiy oynasida *Kompiatorlar* bo'limida dasturiy vosita o'rnatilgan kompyuterda mavjud dasturlash tillari ro'yhati mavjud. Ya'ni foydalanuvchi faqat ushbu dasturlash tillari yordamida dastur tuzish mumkin.[12]

Sahifaning *Yordam* menyusida foydalanuvchida uchraydigan muammolar to'g'risida to'htalib o'tilgan:



Ushbu bo'limdan C++ dasturlash tilida kod yozishdan namuna keltirilgan:



Bu yerda A+B masalani yechishning ikki usuli ya'ni consol va fayl yordamida ishlash ko'rsatilgan.

Pascalda dastur qanday yoziladi? [javob](#)

Sizning dasturingiz kiruvchi ma'lumotlarni standart (konsol orqali) yoki "input.txt" faylidan kiritiladi. Chiquvchi ma'lumotlar esa standart (konsolga) chiqarilishi shart. A+B masalasini yechimi :

```
{SAFETIVE Console}
```

```
uses SysUtils;
```

```
var  
  A, B: Integer;  
begin  
  ReadLn(A);  
  ReadLn(B);  
  WriteLn(A + B);  
  ReadLn;  
end.
```

```
{SAFETIVE Console}
```

```
uses SysUtils;
```

```
var  
  A, B: Integer;  
begin  
  AssignFile(Input, "input.txt");  
  Reset(Input);  
  ReadLn(A, B);  
  WriteLn(A + B);  
end.
```

Huddi shu misolni ikki usuli bilan pascal dasturlash muhitida ishlash ko'rsatilgan.

Javada dastur qanday yoziladi? [javob](#)

Sizning dasturingiz kiruvchi ma'lumotlarni standart (konsol orqali) yoki "input.txt" faylidan kiritilishi mumkin, Lekin chiquvchi ma'lumotlar standart (konsolga) chiqarilishi kerak. A+B masalasini yechimi :

```
import java.io.*;  
import java.util.*;
```

```
class Solver
```

```
{  
    public static void main(String[] args) throws IOException
```

```
{  
    BufferedReader in = new BufferedReader(new InputStreamReader(System.in));  
    StringTokenizer st = new StringTokenizer(in.readLine());  
    int a = Integer.parseInt(st.nextToken());  
    int b = Integer.parseInt(st.nextToken());  
    int s = Integer.parseInt(st.nextToken());  
    int result = a + b;  
    System.out.println(result);  
}
```

```
import java.io.*;  
import java.util.*;
```

```
class Solver
```

```
{  
    public static void main(String[] args) throws IOException
```

```
{  
    BufferedReader in = new BufferedReader(new FileReader("input.txt"));  
    StringTokenizer st = new StringTokenizer(in.readLine());  
    int a = Integer.parseInt(st.nextToken());  
    int b = Integer.parseInt(st.nextToken());  
    int s = Integer.parseInt(st.nextToken());  
    int result = a + b;  
    System.out.println(result);  
}
```

A+B misolni ikki usuli bilan java dasturlash tilida ishlab ko'rsatilgan.

C# .NET da dastur qanday yoziladi? [javob](#)

Sizning dasturingiz kiruvchi ma'lumotlarni standart (konsol orqali) yoki "input.txt" fayli orqali olishi mumkin, Lekin chiquvchi ma'lumotlar standart (konsolga) chiqarilishi kerak. A+B masalasini yechimi :

```
using System;
```

```
class Solver
```

```
{  
    public static void Main(string[] args)  
    {  
        string[] tokens;  
        tokens = Console.ReadLine().Split(" ");  
        int a = Convert.ToInt32(tokens[0]);  
        tokens = Console.ReadLine().Split(" ");  
        int b = Convert.ToInt32(tokens[0]);  
        Console.WriteLine(a + b);  
    }  
}
```

Ushbu misolni C# dasturlash tilida yechilgan kodi ko'rsatilgan.

J# .NET da dastur qanday yoziladi? [javob](#)

Sizning dasturingiz kiruvchi ma'lumotlarni standart(konsole) orqali) yoki "input.txt" fayli orqali olishi mumkin, Lekin chiquvchi ma'lumotlar standart (consolega) chiqarilishi kerak. A+B masalasi yechimi :

```

import System.*;

class solver
{
    public static void Main()
    {
        String[] tokens;
        tokens = Console.ReadLine().Split(new char[] { ' ' });
        int a = Convert.ToInt32(tokens[0]);
        tokens = Console.ReadLine().Split(new char[] { ' ' });
        int b = Convert.ToInt32(tokens[0]);
        Console.WriteLine(a + b);
    }
}

```

J# dasturlash muhitida misollardan biri A+B misolini yechish bo'yicha namuna keltirilgan.

Visual Basic .NET da dastur qanday yoziladi? [javob](#)

Sizning dasturingiz kiruvchi ma'lumotlarni standart(konsole) orqali) yoki "input.txt" fayli orqali olishi mumkin, Lekin chiquvchi ma'lumotlar standart (consolega) chiqarilishi kerak. A+B masalasi yechimi :

```

Imports System

Module solver

Sub Main()
    Dim tokens As String()
    tokens = Console.ReadLine().Split(New Char() { " " })
    Dim a As Integer = Convert.ToInt32(tokens(0))
    tokens = Console.ReadLine().Split(New Char() { " " })
    Dim b As Integer = Convert.ToInt32(tokens(0))
    Console.WriteLine(a + b)
End Sub

End Module

```

Visual Basic .NET da A+B misoli ishlab ko'rsatilgan.

Bunda tashqari foydalanuvchi tekshirish uchun yuborgan dasturidan quyidagi natijalar olish mumkin:

Qanday natija olish mumkin? javob	
Accepted!	• Dastur barcha testlardan muvaffaqiyatli o'tdi.
Compilation Error	• Dastur matnida sintaktik xato bor. • Xato kompilator / til tanlangan.
Wrong Answer	• Dastur qandaydir testdan o'tmadi.
Presentation Error	• Dastur xato formatdagi ma'lumotlarni berdi.
Runtime Error	• Tekshirishda xatolik sodir bo'ldi.
Time Limit	• Dastur belgilangan vaqtdan ko'p vaqt sarfladi.
Memory Limit	• Dastur belgilangan xotiradan ko'p joy sarfladi.
Security Violation	• Dastur taqiqlangan funksiyani chaqirdi.
Waiting	• Server hozir boshqa dasturni tekshirmoqda.
Waiting (C)	• Siz tanlagan kompilator serverda topilmadi.
Checking	• Dastur hozir tekshirilmoqda.

2.2.Dasturiy vosita tasnifi

Dasturiy vosita foydalanuvchi interfeysi

Dasturiy vositaga foydalanuvchi kompyuteridagi brouzeri orqali lokal yoki global tarmoq orqali dasturiy vosita joylashgan kopyuterning IP Adressini yozish orqali kiriladi.



Ushbu dasturiy vosita ikkita tilda ya'ni o'zbek va rus tillarida mavjud bo'lib, foydalanuvchi asosiy oynaning *saytga a'zo bo'lish* tugmasini bosish natijasida quyidagi oyna hosil bo'ladi:

Ism*		рекомендуется имя и фамилия
Parol*		
Takror parol*		
Shaxsiy e-mail		Faqat administrator va siz uchun sizning sahifangizda ko'rinmaydi.
Ochiq e-mail		Sizning sahifangizda ko'rsatiladi.
Shaxsiy ICQ		Faqat administrator va siz uchun sizning sahifangizda ko'rinmaydi.
Ochiq ICQ		Sizning sahifangizda ko'rsatiladi.
Shahar, davlat (rus)		
Shahar, davlat (o'z)		
O'qish joyi (rus)		
O'qish joyi (o'z)		
Tugilgan k		
		Rasmdagi sonni kiriting
OK Orqaga		

Ushbu sahifa foydalanuvchi tomonidan kiritiladigan ma'lumotlar uchun. Ma'lumotlar kiritlgandan so'ng foydalanuvchilar bo'limida yagni foydalanuvchi qo'shiladi. Foydalanuvchi login, parolini kiritish orqali aktivlashadi va ekranda foydalanuvchi to'g'risida quyidagi ma'lumotlar chiqadi[13]:

Sanjar
Mening sahifam
Yechilgan mas.: 0
Barcha urinishlar: 0
Omadsiz urinishlar: 0
Chiqish

Foydalanuvchi *misollar* menyusi orqali dasturiy vositada mavjud bo'lgan masalalar ko'rish va ularni ishlash imkoniyati mavjud:

Misollar		
nomi	Masala..	Forum
Buxvalar	1 masala	Muhok..
Qidirish	1 masala	Muhok..
Sonlar nazariyasi	5 masala	Muhok..
Sohlar	1 masala	Muhok..
Saratash	1 masala	Muhok..
Arifmetika	7 masala	Muhok..
Rekursiya	0 masala	Muhok..
Geometriya	1 masala	Muhok..
Etila	1 masala	Muhok..

Bu yerda masalalar mavzu bo'yicha guruhlariga ajratilgan va har bir guruh masalalar soni va muhokamasi bo'limlari mavjud. Ajratilgan guruhlardan biriga kirish natijasida unda mavjud masalalar va ushbu masalalar to'g'risida qisqacha

ma'lumotlar va ushbu masalalar muhokamasi mavjud oynaga o'tiladi. Ushbu masalalardan birini tanlash oqibatida quyidagi sahifa hosil bo'ladi:

Kvadrat

Vaqt 1000/1000/1000/1000 ms. Xotira 65000/65000/65000/65000 Kb.

5 soniga qoldiqsiz bo'linadigan butun n soni berilgan. Bu sonning kvadratini toping.

Kiruvchi ma'lumotlar
Butun n soni berilgan, $0 \leq n \leq 30370004999$, n mod 5=0 bo'lishi kafolatlanadi(mod qoldiq olish amali).

Chiquvchi ma'lumotlar
Sonning kvadratini chiqaring.

Nº	Kiruvchi ma'lumotlar	Chiquvchi ma'lumotlar
1	15	225

Jo'natish

Masala: Kvadrat

Dastur matni

Til: C++

[Fayl jo'natish](#)

[Dastur qanday yoziladi?](#)

[Qanday natija olish mumkin?](#)

[Jo'n.](#)

Ushbu bo'limda *Kvadrat* nomli masala va unga testlardan namunalar berilgan. Jo'natish qismida dastur kodi va yozilgan dasturlash muhiti va jo'natish tugmasi berilgan. Bundan tashqari dasturni fayl orqali yuborish va dastur yozish uchun namunalar keltirilgan. Berilgan masala uchun dastur kodini yozib natijasi ekranda hosil bo'ladi[14]:

Urinishlarim

Wrong Answer	test 1	C++ ochiq kod	info ▾
Wrong Answer	test 7	C++ ochiq kod	info ▴
			kod ▴

```

1 #include <iostream>
2 using namespace std;
3 int main(){
4     int a;
5     cin>>a;
6     cout<<a*a;
7     return 0;
8 }

```

Yuqoridagi misolda sonning kvadratini aniqlash masalasi qaralgan. C++ dasturlash tilida yozilgan dastur berilgan testlarning yettinchisidan o'tmaganligini anglatuvchi xabar chiqarildi. Chinki misolda sonning chegarasi haqida ma'lumot berilgan. Misolda sonning chegarasi kichikligi sababli undan kattaroq songa hatolik olib keladi. Natijada dastur noto'g'ri yozilganligi haqida ma'lumot olindi[15][16].

Java dasturlash tilida biror masalani to'g'ri ishlagandan keyingi sahifa quyidagicha:

Urinishlarim

Accepted!

Java ochiq kod

info

kod

```

1 import java.util.Scanner;
2 public class solver
3 {
4     public static void main(String[] args)
5     {
6         Scanner in=new Scanner(System.in);
7         int s,v,t;
8         s=in.nextInt();
9         v=in.nextInt();
10        t=in.nextInt();
11        if(v*t>=s)System.out.print("Qizil darsga ulguradi");
12        else System.out.print("Yana bitta NB");
13    }
14 }
```

Foydalanuvchi administrator o'tkazmoqchi bo'lgan contestlar vaqtini ko'rish va unga qatnashish xuquqiga ega. O'tib ketgan contestlarning natijasini ko'rish mumkin. Ushbu natijalarni *Contest* menyusidan o'tkazilgan contestlarning biriga kirib *monitor* tugmasini bosib kirish mumkin.

Contest#1							
Contest tugadi!							
03:54 Enter Accepted! D. Yashin ajdab							
#	User	A	B	C	D	E	
		Darslar	Solent	Qol va afvlar	Tashv solent	Solent darslar...	
1	Abu Amr	+	+	+2	-1	+	4
2	Sade	+	+1	+1	03:53	03:49	4
3	Amir	+1	+1	-4	-3	+1	3
4	Abdusol	+1	+1	-4	03:06	03:19	3
5	Amir	+1	+4	-4	03:13	03:26	2
6	Amir	+1	+4	-4	03:13	03:26	2
7	Sherzod	+	-1	-1			1
8	Muhammad	+	-1	-1			1
9	Enter	+	-1	-1			1
	Unosha	12	14	13	11	7	
	Accepted	6	6	2	1	3	
	Rejected	4	9	11	10	2	

Ekkranda o'tkazilgan contest#1 bo'yicha natijalarni ko'rish mumkin. Ushbu reyting acm qoidasi asosida ishlaydi, ya'ni ishtirokchilar masalalar soni kamayish bo'yicha tartiblanadi, agar masalalar soni teng bo'lsa jarima ballari o'sish tartibida tartiblanadi. Jarima bali sifatida foydalanuvchi masalani contest boshlangan vaqtga nisbatan ishlagan vaqti va har bir noto'g'ri urinishi uchun 20

jarima bali birgalikda qo'shib hisoblanadi. Agar masalani bir necha urinishda o'tkaza olmasa u jarima baliga qo'shilmaydi[17][18].

Dasturiy vosita administrator interfeysi

Dasturiy vosita Administrator yoki Administrator huquqi berilgan foydalanuvchi kirish imkoniyatiga ega:



User	Shahar,davlat	O'qish joyi	AC	RI
Mansurbek			70	42
Amir	O'zbekiston,Samarqand	TATU SP	4	10
Sardor			4	2
Abdurahimov	O'zbekiston	TATU SP	4	3
Abdusaid			3	11
Auliddin			2	5
usbek			2	3
Shernobek			1	1
Muhammad			1	0
Hamidulloxo Mansurbek			1	0
Emir			1	1
Sanjar	O'zbekiston, Samar...	TATU Samarqand filial	1	9
Admin			0	0

Administrator huquqini beradigan foydalanuvchini faqat administrator ro'yhatdan o'tkazadi:




Shu joydan foydalanuvchi administrator huquqiga egaligi yoki foydalanuvchi huquqiga ega ekanligini aniqlash mumkin.

Administratorda foydalanuvchilar tomonidan yuborilgan barcha kodlarni ko'rish, yangi masala tuzish, tuzilgan masalaga test tuzish va ushbu testlarga cheker(yuborilgan kodni maxsus testlar bilan solishtirib testlovchi dsatur) yozish imkoniyatlari mavjud. Foydalanuvchilar bo'limidan biror foydalanuvchining barcha yuborgan kodlarini ko'rish imkoniyati mavjud:

Urinishlar					
May.28.2015 06:24:31 PM	C. Qizil va afsungar	Wrong Answer	test 1	C++	Info
May.28.2015 06:21:25 PM	C. Qizil va afsungar	Wrong Answer	test 1	C++	Info
May.28.2015 06:17:39 PM	C. Qizil va afsungar	Wrong Answer	test 1	C++	Info
May.28.2015 06:08:29 PM	Eng katta son	Accepted!		C++	Info
May.28.2015 06:07:09 PM	Eng katta son	Wrong Answer	test 5	C++	Info
May.28.2015 06:05:51 PM	Eng katta son	Wrong Answer	test 7	C++	Info
May.28.2015 06:04:16 PM	Eng katta son	Runtime Error	test 7	C++	Info
May.16.2015 12:59:46 PM	D. Yashilni qidirish	Wrong Answer	test 1	C++	Info
May.16.2015 12:57:03 PM	D. Yashilni qidirish	Time Limit	test 8	C++	Info
May.16.2015 12:14:22 PM	E. Sariq dastur tuzmoqda.	Accepted!		C++	Info
May.16.2015 12:01:38 PM	D. Yashilni qidirish	Time Limit	test 7	C++	Info
May.16.2015 11:59:15 AM	E. Sariq dastur tuzmoqda.	Wrong Answer	test 1	C++	Info
May.16.2015 11:58:27 AM	D. Yashilni qidirish	Time Limit	test 5	C++	Info
May.16.2015 09:46:48 AM	C. Qizil va afsungar	Wrong Answer	test 11	C++	Info
May.16.2015 09:44:54 AM	C. Qizil va afsungar	Runtime Error	test 1	C++	Info
May.16.2015 09:41:26 AM	C. Qizil va afsungar	Wrong Answer	test 11	C++	Info
May.16.2015 09:38:39 AM	C. Qizil va afsungar	Wrong Answer	test 4	C++	Info
May.16.2015 09:30:26 AM	D. Yashilni qidirish	Runtime Error	test 1	C++	Info
May.16.2015 09:25:44 AM	B. Sayohat	Accepted!		C++	Info
May.16.2015 09:22:02 AM	B. Sayohat	Wrong Answer	test 3	C++	Info
May.16.2015 09:15:20 AM	A. Darsga borish	Accepted!		C++	Info
May.16.2015 09:14:24 AM	A. Darsga borish	Wrong Answer	test 2	C++	Info

Ekranda Anvar ismli foydalanuvchining yuborgan kodlari, vaqti, to'g'ri yoki noto'g'riligi nechanchi testdan qaytganligi, qaysi dasturlash tilida yuborganligi va *info* tugmasi mavjud. *Info* tugmasini bosish natijasida Administrator ushbu foydalanuvchining yuborgan kodini va qaysi testdan qaytganligi yuborilgan dastur chiqargan natija va to'g'ri javob hosil bo'ladi:

```

May 16, 2015 09:40:40 AM C. Qitil va afsonqar Wrong Answer test 11 C++ info
#include<iostream>
#include<stdio.h>
using namespace std;
int main ()
{
    long long int n,i,j,k,w=0,w1;
    scanf("%d",&n);
    if(n==0)
    {
        printf("%d",0);
        return 0;
    }
    for(i=1;i<=n;i++)
    {
        scanf("%d",&w);
        if(w==1)
        {
            w1=w1+1;
        }
    }
    printf("%d",w1);
}

```

May 16, 2015 09:44:54 AM C. Qitil va afsonqar Runtime Error test 1 C++ info

May 16, 2015 09:41:26 AM C. Qitil va afsonqar Wrong Answer test 11 C++ info

May 16, 2015 09:38:20 AM C. Qitil va afsonqar Wrong Answer test 5 C++ info

May 16, 2015 09:30:36 AM D. Yashirov qidirish Runtime Error test 1 C++ info

User bo'limida foydalanuvchilar reytingi mavjud. Foydalanuvchilarni ko'p ishlagan masalalari bo'yicha saralab ekranga chiqaradi:

Userlarni saralash: [yechilgan masalalar bo'y.] [Registr mal.] [alfbo bo'yicha.]

User	Shahar,davlat	O'qish joyi	AC	RJ
Mansurbek			70	42
Anvar	O'zbekiston,Samarqand	TATU SF	4	18
Sardor			4	2
Abror Amirqulov	O'zbekiston	TATU SF	4	3
Abduvohid			3	11
Asliddin			2	5
oybek			2	3
Sherzodbek			1	1
Muhammad			1	0
Narzullayev Mansurbek			1	0
Enter			1	1
Sanjar	O'zbekiston, Samar...	TATU Samarqand filiali	1	9
Admin			0	0
Qurbonqoy_5i			0	0

Bundan tashqari foydalanuvchilari ro'yhatdan o'tgan muddati bo'yicha yoki alifbo bo'yicha saralashimkoniyati mavjud.

3.4. Dasturiy vosita masalalar arxivi

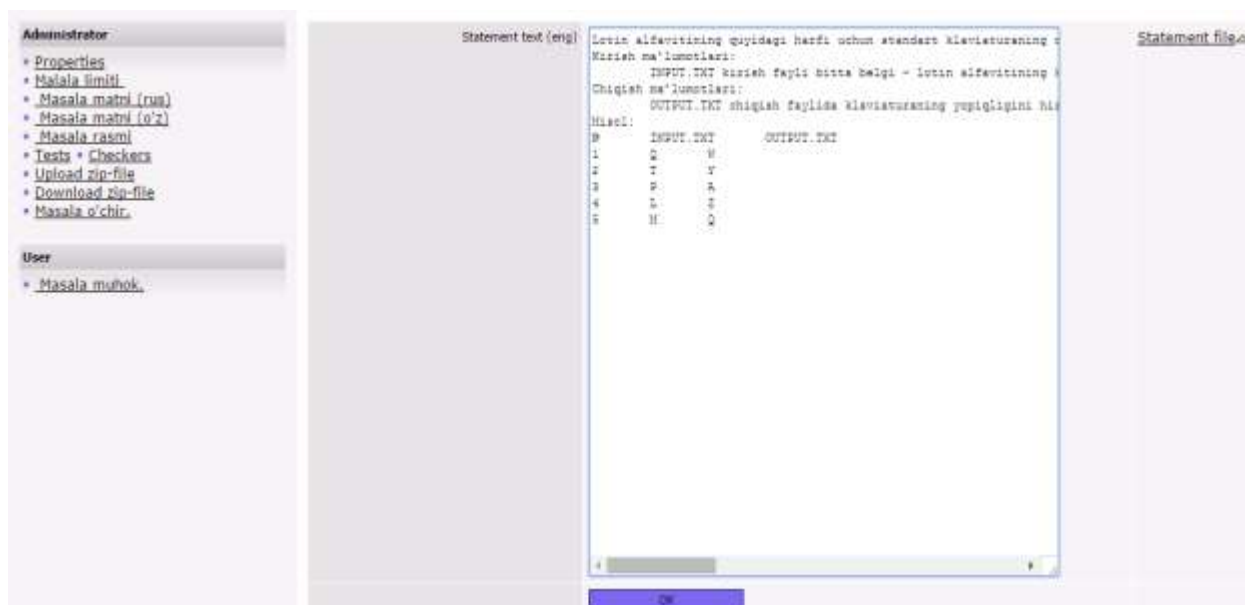
Administrator tomonidan yangi masala qo'shish imkoniyati mavjud:

Nomi (rus)*		recommended with a capital letter
Nomi (o'z)*		recommended with a capital letter
Qisqacha (rus)		essential phrases from statement
Qisqacha (o'z)		essential phrases from statement
Note (rus)		author, year, source, internet page
Note (eng)		author, year, source, internet page
Vaqt (ms)	2000 4000 4000 4000	C++, Pascal Java .NET langs others
Xotira (Kb)	65000 65000 65000 65000	C++, Pascal Java .NET langs others
AutoLimits	☐ Take limits from admin's solutions. More »	
Section*	Fizika ▼	
Volume	outside any volume ▼	
Sbornik nomeri		
Contest	outside any contest ▼	
Contest nomeri		
Daraja	Not set ▼	
OK Orqaga		

Dastur interfeysi ikki tilda yaratilgani sabab bo'limda masala nomi, masala to'g'risida qisqacha ma'lumot, masalani qo'shgan foydalanuvchi ismi sharifi ikki tilda rus va o'zbek tillarida kiritilishi so'ralgan. Bundan tashqari *vaqt* bo'limida foydalanuvchi yuborgan dasturning optimalligini tekshiradi. Ya'ni dastur kodini testlash jarayonida berilgan vaqt limitidan oshib ketgan dasturni *timelimit* hatoligini beradi. *Xotira* bo'limida ushbu masala uchun tuzilgan dastur xotiradan egallashi lozim bo'lgan maksimal xotira ko'rsatiladi. Masala tegishli bo'lgan bo'lim, contest va masalaning qiyinlik darajasi to'g'risida ma'lumotlar kiritiladi. Ushbu ma'lumotlar to'ldirilgandan so'ng quyidagi sahifaga o'tiladi[19][20]:



Ushbu sahifadan *masala matni* bo'limiga o'tiladi:



Masalaga rasm qo'shish imkoniyati ham mavjud. *Testlar va Checkerlar* bo'imi to'ldirilgandan so'ng masala aktivlashadi. *Upload zip-file* bo'limi yordamida masalani barcha holatlarini to'ldirib masala qo'shish imkoniyati mavjud. *Download zip-file* bo'limi yordamida masalani hamma holatlari bilan birgalikda xotiraga saqlab olish mumkin. Eski masalani o'chirish imkoniyati ham mavjud[21].

2.3. Tuzilgan masalani to'g'riligini tekshirish uchun testlar yaratish va testlash

Testlar bo'limida tuzilgan masala uchun test tuzish bo'limi hisoblanadi:

New test pair	
Test number	 if not set, then extracted from input.txt name
input.txt	Выберите файл Файл не выбран
pattern.txt	Выберите файл Файл не выбран optional file
Create	

Bu sahifada ikkita fayl input.txt va pattern.txt fayllarini kiritish so'raladi. input.txt fayli tuzilgan masalani testlash fayli hisoblanadi. Pattern.txt fayli ushbu kirish testida masalaning to'g'ri javobi yozilgan fayl beriladi[27].



Ushbu saqlangan fayllarni dasturga kiritiladi:

1	input.txt	pattern.txt	delete	ТЕКСТ
	Q	W		
2	input.txt	pattern.txt	delete	ТЕКСТ
	E	A		
3	input.txt	pattern.txt	delete	ТЕКСТ
	E	E		
4	input.txt	pattern.txt	delete	ТЕКСТ
	X	C		

Yuqoridagi sahifada berilgan masala uchun 4 ta test berilgan, ya'ni kirish testlari va natijaning to'g'riligi tekshiruvchi chiqish testlari. Foydalanuvchi tomonidan yuborilgan dastur kodiga kirish testini berib, olingan natijani pattern.txt testi bilan solishtiriladi. Agar foydalanuvchi dasturidan olingan natija chiqish testi bilan to'g'ri kelmasa unga ushbu dastur ushbu test raqamida noto'g'ri javob berilganligi to'g'risida habar beriladi. Ushbu holat checker qismida bajariladi:

Checkers				
minut	Approved!	Pas checker text	info△	
	delete		code△	log▽
1	program checker;			
2				
3	{ \$APPTYPE CONSOLE }			
4				
5	uses			
6	SysUtils,			
7	{ \$IFDEF MSWINDOWS }			
8	unJudge17 in '..\utils\unJudge17.pas';			
9	{ \$ELSE }			
10	unJudge17 in '..\utils\unJudge17.pas';			
11	{ \$ENDIF }			
12				
13	var			
14	o,p,i:char;			
15	begin			
16	InputTxt.ReadChar(i);			
17	OutputTxt.Readchar(o);			
18	OutputTxt.MustBeEof;			
19	PatternTxt.ReadChar(p);			
20	if o <> p then VRejectWA(Format('%s%s=%s bu xato,to''g''ri javob %s%s=%s',[i,i,o,i,i,p			
21				
22	));			
23	VAccept(Format('OK %s%s=%s',[i,i,o]));			
24	end.			
07:30:01 AM	Bad Checker or Input	Pas checker text	info▽	

Ya'ni checker masala uchun yaratgan testlaringiz bilan solishtirish uchun tuzilgan. Pascal dasturlash tilida tuzilgan unjudge17.pas checker fayli mavjud. Ushbu checker faylidan foydalangan holda dastur Output.txt.Readchar() procedurasi yordamida foydalanuvchi yuborgan dastur natijasini o'qitadi va Pattern.txt.ReadChar() procedurasi yordamida to'g'ri javobni o'qitadi va ularni bir – biri bilan solishtiradi. Agarda adiministrator tomonidan tuzilgan testlardan birortasida hatolik sodir bo'lsa foydalanuvchiga test nomeri bo'yicha hatolik xabari kelib tushadi. Barcha testlardan muvoffaqiyatli o'tgan dasturda foydalanuvchiga dastur to'g'ri ishlanganligi to'g'risida ma'lumot kelib tushadi. Masalaning yechimi bir nechta bo'lishi mumkin. Bunday masalalarni yechishda dasturiy ta'minot yordam ko'rsata oladi[28]. Bunday xildagi masalalarni yechishda checker dasturiy qismi c++ dasturlash tilida yasalgan bo'lib quyida biror bir masalaning natijasi bilan tanishib chiqamiz:

Contest > Contest#2 > masala:

D. Omadli raqam

Vaqt 1000/1000/1000/1000 ms. Xotira 65000/65000/65000/65000 Kb.

Vashi bo'sh vaqtlarida omadli telefon raqamlariga telefon qilib turadi. Omadli telefon raqam deb dastlabki to'rt raqami yig'indisi oxirgi to'rt raqami yig'indisiga teng bo'lgan songa aytiladi. **Vashi** endi raqamlari yig'indisi k ga teng bo'lgan omadli telefon raqamiga telefon qilmoqchi. Bu raqamni topishda **Vashi**ga yordam bering.



Kiruvchi ma'lumotlar
 Birinchi qatorda k butun soni berilgan. $2 \leq k \leq 63$.

Chiquvchi ma'lumotlar
 Raqamlari yig'indisi k ga teng bo'lgan omadli telefon raqamini chiqaring. Bu telefon raqami 7 xonadan iborat bo'lishi kerak va bunday son mavjudligi kafolatlanadi. Agar yechim yagona bo'lmasa ulardan istalganini chiqaring.

Eslatma
 Hosil bo'lgan omadli telefon raqami 0 bilan boshlanmasligi kerak.

№	Kiruvchi ma'lumotlar	Chiquvchi ma'lumotlar
1	10	11004012
2	10	11004012

Urinishlarim / Participants / Accepted urinishlar / Urinishlar

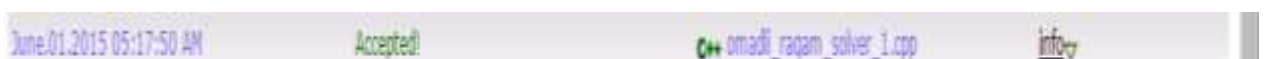
Yechganlar

✓	Gideon	1	0
✓	Mansurbek	1	0
✗	Sirojiddin	0	4

Ushbu masalaning yechimi bir necha hil bo'lishi mumkin. Ko'pchilik avtomatlashtirilgan testlovchi dasturiy ta'minotlar bunday vazifani bajara olmaydi. Ushbu dasturiy ta'minot foydalanuvchilar yuborgan barcha to'g'ri yechimlarni qabul qila oladi. Ushbu masalaning checker fayli bilan tanishib chiqamiz[29]:

```
#include<iostream>
#include<math.h>
#include<algorithm>
#include<fstream>
using namespace std;
void __AC()
{
    exit(0xAC);
}
```

Ushbu funksiya foydalanuvchi tomonidan yuborilgan masala yechimi to'g'riligini qaytaruvchi funksiya hisoblanadi:



```
void __WA()
{
    exit(0xAB);
}
```

minut Wrong Answer test 1 C++ ochiq kod info

Ushbu funksiya foydalanuvchi tomonidan yuborilgan masala yechimi noto'g'riligini qaytaruvchi funksiya hisoblanadi. Agarda dastur hato natija qaytargan holda Administratorga uning xatosi haqida report.txt fayliga ma'lumot beradi:

```
----- report.txt -----
Javob xato - 7112117
To'g'ri javob - 6001006
----- input.txt -----
13
----- output.txt -----
7112117
----- pattern.txt -----
6001006
```

```
void __PE()
{
    exit(0xAA);
}
```

minut Presentation Error test 1 C++ ochiq kod info

Ushbu funksiya foydalanuvchi tomonidan yuborilgan masala kodi hato formatdagi ma'lumotlarni qaytaruvchi funksiya hisoblanadi;

```
void __FAIL()
{
    exit(0xA3);
}
```

Ushbu funksiya administrator tomonidan tuzilgan checker yoki testlarda hatolik sodir bo'lganligi haqida habar beruvchi funksiya hisoblanadi;

```
bool IsDigit(char a)
{
    return ('0'<=a && a<='9');
}
```

Ushbu funksiya berilgan belgining raqam yoki raqam emasligini aniqlaydigan funksiya hisoblanadi;

```
bool IsLongLong(string s)
{
    if(s.length()==1 && !IsDigit(s[0]))return false;
    if(s[0]!='-'&& !IsDigit(s[0]))return false;
    for(int i=1;i<s.length();i++)
    {
        if(!IsDigit(s[i]))return false;
    }
    return true;
}
```

Ushbu funksiya berilgan satrning butun turga tegishliligi aniqlaydigan funksiya hisoblanadi;

```
__int64 StrToInt(string s)
{
    __int64 k=1,Sum=0;
    if(s[0]=='-')k*=-1;
    for(int i=0;i<s.length();i++)
    {
        if(s[i]=='-')continue;
        Sum=Sum*10+s[i]-'0';
    }
    Sum*=k;
}
```

```

        return Sum;
    }

```

Ushbu funksiya string turida berilgan qiymatni butun turga o'tkazuvchi funksiya hisoblanadi;

```

bool IsDouble(string s)
{
    int count_tochka=0;
    if(s=="-.")return false;
    if(s.length()==1 && !IsDigit(s[0]))return false;
    if(s[0]!='-'&&s[0]!='.'&&!IsDigit(s[0]))return
false;
    for(int i=0;i<s.length();i++)
    {
        if(s[i]=='.')count_tochka++;
        if(i&&!IsDigit(s[i])&&s[i]!='.')return false;
    }
    if(count_tochka>1)return false;
    return true;
}

```

Ushbu funksiya berilgan satrning xaqiqiy turga tegishliligini aniqlaydigan funksiya hisoblanadi;

```

int main ()
{
    string S,P;
    int t,v,k;
    ifstream in("input.txt"); // Input.txt faylini
testni olish uchun ochish
    ifstream pat("pattern.txt");// pattern.txt faylini
to'g'ri javobni olish //uchun ochish

```

```
ofstream rep("report.txt");// report.txt faylini
yuborilgan dastur xaqida ma'lumot berish uchun ochish
    freopen("output.txt","r",stdin); //output.txt
faylini Yuborilgan dastur natijasini olish uchun
ochish
```

```
    in>>k;
    if(!(cin>>S))
    {
        rep<<"Javob chiqarilmadi";
        rep.close();
        __PE();
    }
    int n=k;
    string s="0000000";
    for(int i=0;i<3;i++)
    {
        if(n>18)
        {
            s[i]='9';
            s[s.length()-1-i]='9';
            n-=18;
        }
        else if(n>1)
        {
            s[i]=char(n/2+'0');
            s[s.length()-1-i]=char(n/2+'0');
            n-=(2*(n/2));
        }
        else
        {
```

```

        s[i]=char('0');
        s[s.length()-1-i]=char('0');
    }
}
if(n)
{
    s[3]=char(n+'0');
}
if(cin>>S)
{
    rep<<"Javob yagona butun son bo'lishi kerak";
//javob yagona bo'lmasa
    rep.close();
    __PE();
}
if(S.length()!=7||S[0]=='0')
{
    rep<<"Javob xato -      "<<S<<"\n"<<"To'g'ri
javob -      "<<s;
    rep.close(); // Berilgan shartlarni
qanoatlantirmasa
    __WA();
}
for(int i=0;i<S.length();i++)
{
    if(S[i]>'9'||S[i]<'0')
    {
        rep<<"Javob xato -      "<<S<<"\n"<<"To'g'ri
javob -      "<<s;

```

```

        rep.close();//Berilgan shartlarni
qanoatlantirmasa
        __WA();
    }
}
int h=0,a1=0,a2=0;
for(int i=0;i<S.length();i++)h+=S[i]-'0';
for(int i=0;i<4;i++)a1+=S[i]-'0';
for(int i=3;i<7;i++)a2+=S[i]-'0';
if(h!=k||a1!=a2)
{
    rep<<"Javob xato -      "<<S<<"\n"<<"To'g'ri
javob -      "<<s;
    rep.close(); //Berilgan shartlarni
qanoatlantirmasa
    __WA();
}
__AC(); //barcha shartlarni qanoatlantirsa
rep<<"Javob to'g'ri";
rep.close();
}
[29][30][31]

```

2.4. Kompyuter bilan ishlashda hayotiy faoliyat xavfsizligi qoidolari va talablari.

Kompyuter bilan ishlash davomida bir qator faoliyat bilan bog'liq fiziologik jarayonlar sodir bo'ladi va inson organizmiga turli darajada ta'sir ko'rsatadi.

Bu kabi surunkali ta'sirlar va xavfli holatlar muxandis-dasturlarni ish faoliyatiga ta'sir ko'rsatib kasb kasaliklarini keltirib chiqarishi tasdiqlangan.

Kompyuter lar joylashtirilgan xonalar va dasturlovchi o'rni yong'in xavfsizligi talablari bo'yicha «D» toifaga mansub bo'lib, tez yonuvchi materiallar kam hisoblanadi[22].

Elektr hisoblash mashinalari va aloqa vositalari joylashtirilgan xonalarda yong'in xavfi mavjud bo'lib, bu holat katta miqdorda tez eriydigan engil materiallar. Bu xonalarda yong'in asosan yonuvchan moddalarning oksidlanishi, issiqlik ta'sirida erishi natijasida sodir bo'ladi. Mavjud xonalarda o'rnatilgan akustik va estetik talablarga mos keluvchi eshik, pol, kabellar izolyasiyasi va boshqalar yonuvchi komponentlar toifasiga kiradi.

Yong'inga qarshi kurash bu kompleks tashkiliy va texnik tadbirlar bo'lib, xodimlarni xavfsizligini ta'minlash, yong'inni samarali o'chirish uni tez tarqalishini oldini olishdan iborat. Uchqun chiqaruvchi manbalar esa, elektron sxemalar, elektr ta'minot tarmoqlari, turli elektron uskunalarning nosozligi, qisqa tutashuv, er bilan notug'ri biriktirish, ulashdagi nosozliklar hisoblanadi[23].

Zamonaviy Kompyuter lar va EHMlarida elektron sxemalar va detallar zich joylashganligi va kabellar hamda ulash simlari yaqinligi sababli bu xavf ancha yuqoridir. Bulardan elektr tokini oqishi himoya qavatini erishiga olib keladi. Kompyuter larda bu kabi ortiqcha issiqlik maxsus shamolatish bloki orqali chiqarib yuboriladi. Bu blokni ishdan chiqishi esa juda xavflidir. Bunday hollarda xavfsizlik choralarini ko'rish va qo'shimcha shamollatish tizimini tashkil etish maqsadga muvofiqdir.

Ma'lumki, yong'in holatini Kompyuter choralarida qo'llanilgan qurilish materiallarining yong'inga chidamligini, umumiy mustahkamlikni ta'minlaydi. Elektron uskunalarni yuqori narxini e'tiborga olgan holda bu turdagi xonalarni 1 yoki 2 toifali yong'inga chidamlilik deb darajalash mumkin[24].

Yong'in xavfsizligi qoidalariga binoan EHM joylashtirilgan xonalar, ko'paytirish qiyofa ko'chirish uskunalari atrofida tutunli xabar berish moslamalari o'rnatiladi. Yong'in boshlangan holatda plastmassalarni tutashi natijasida zaharli is gazi ajralib chiqadi va ishchilarni bug'ishi mumkin. Shu sababli, belgilangan talablarga mos keladigan o'chirish usullari va nazariyasi- dan foydalanish

maqsadidga muvofiqdir. Olib borilayotgan tadqiqotchilar va tahlillar asosida aynan shu holatga mos keluvchi usullar tanlanadi. Ish o'rnini samarali tashkil etishda asosiy e'tibor elektromagnit va elektrostatik maydonlar ta'sirini kamaytirishga qaratilishi darkor. Bugungi kunda ish o'rnini ergonomik talablar asosida tashkil etish, komfort ish sharoitini tashkil etish dolzarb sosial masaladir. Olingan tadqiqotlar asosida ish stolining optimal o'lchamlari, qo'lay konfiguratsiya, ish o'rnini to'g'ri tashkil etishda asosiy manbadir.

Belgilangan talablarga javob beradigan sharoit yaratish, ish sifatini va unumdorligini oshirish, dastur ishini yaxshilash, nuqsonsiz ishlashni ta'minlash imkonini beradi.

Kompyuter bilan ishlash davomida insonlarning ko'rish organlarida yuqori darajada kuchlanish holati sodir bo'ladi.

Turli o'lchamdagi va hajmdagi grafik materiallar, kichik o'lchamlar, hajmini ortishi natijasida ko'zda surunkali jiddiy kuchlanish yoki zo'riqish sodir bo'ladi. Bu kabi salbiy holatlar ko'zni ko'rish qobiliyatini pasayishiga olib keladi.

Belgilangan me'yorlarga binoan foydalanuvchi operator ko'zi bilan ekran oralig'idagi masofa 600-700 mm bo'lishi lozim, Ayrim hollarda harf va raqamlar kattaligini e'tiborga olgan holda bu masofani 500 mm bo'lishiga ruxsat beriladi.

Ko'zni zo'riqishini kamaytirish maqsadida har 1 soatda 15 minutlik tanaffus qilish tavsiya etiladi.

Bugungi kunda bir qator olimlar tomonidan Kompyuter monitorlaridan tushayotgan turli nurlarni inson organizmiga ta'siri va ma'lum ionlanish sodir bo'lishi tasdiqdangan. Lekin, bu masalada to'laqonli belgilar mavjud emas.

Elektromagnit ionlashgan nurlarning monitordan tushadigan miqdori qo'yidagi jadvalda keltirilgan[25].

Ma'lumki, tegishli me'yorlarga asosan ish joyida Kompyuter monitoridan ajraladigan maksimal rengen nurlarning maksimal miqdori 10 mk ber/ch va ultrabinafsha va infraqizil nurlarning bu kabi miqdori esa 10-100 mVt/m² bo'lishi aniqlangan.

XULOSA VA TAKLIFLAR

Bitiruv malakaviy ishida qaralgan masalalardan quyidagi natijalar va xulosalarga kelimiz:

- Dasturni teslovchi avtomatlashtirilgan tizimlar ishlash metodlari o'rganildi;
- Dasturlash bo'yicha olimpiadalarni o'tkazishga ko'maklashuvchi dasturiy vositani yaratish va misollar yechish uchun ma'lumotlar to'plandi;
- Ma'lumotlar to'planib avtomatlashtirilgan testlovchi tizim yaratildi;
- Yaratilgan dasturiy vosita masalalar bilan boyitildi;
- Tuzilgan masalalar uchun maxsus testlar tuzildi;
- Yuborilgan dasturni testlash uchun maxsus chekkerlar yaratildi;
- ishlab chiqilgan dasturiy vositaning tasnifi va undan foydalanish tartibi keltirildi;
- Fan bo'yicha ishlab chiqilgan dasturiy vositadan axborot texnologiyalar va texnika sohasidagi oliy o'quv yurtlarida dasturlash bo'yicha fan olimpiadalarni o'tkazishda foydalanish mumkin.

ADABIYOTLAR

1. Karimov I.A. Bosh maqsadimiz – keng ko’lamli islohatlar va modernizasiya yo’lini qat’iyat bilan davom ettirish: 2012-yilda mamlakatimizni ijtimoiy-iqtisodiy rivojlantirish yakunlari hamda 2013-yilga mo’ljallangan iqtisodiy dasturning eng muhim ustivor yo’nalishlariga bag’ishlangan O’zbekiston Respublikasi VMning majlisidagi ma’ruza. –T.: O’zbekiston, 2013. -64 b.
2. Karimov I. A. «Zamonaviy axborot-kommunikasiya texnologiyalarini yanada joriy etish va rivojlantirish chora-tadbirlari to’g’risida» PQ-1730 sonli qarori. –T.: O’zbekiston, 2012
3. Karimov I. A. “Joylarda kompyuterlashtirish va axborot-kommunikasiya texnologiyalarini yanada rivojlantirish uchun shart-sharoitlar yaratish chora-tadbirlari to’g’risida”gi qarori. –T.: O’zbekiston, 2012
4. Karimov I.A. Inson, uning huquq va erkinliklari – oliy qadriyat. – T. 14. – Toshkent: O’zbekiston, 2006. – 280 b.
5. Barkamol avlod orzusi // Tuzuvchilar: Sh. Qurbonov, R.Axlidinov, H.Saidov.–Toshkent: Sharq, 1999.–205 b.
6. Oliy ta’lim me’yoriy-huquqiy hujjatlar to’plami. 1-2-tom. Toshkent, «Niso Poligraf» ShK bosmaxonasi, 2013.
7. Axborot va axborotlashtirishga oid normativ-huquqiy hujjatlar to’plami. //To’plovchilar: A.I.O’ralov, M.I.Ishbekov, D.S.Sa’dullayev. –Toshkent, “Adolat” nashriyoti, 2008. -464 b.
8. Karimov I.A. O’zbekiston iqtisodiy islohatlarni chuqurlashtirish yo’lida. -Toshkent.: O’zbekiston. 1995.
9. Aripov M. Informatika va hisoblash texnikasi asoslari.- Toshkent.: Universitet, 2000. -690 b.
- 10.Sh. F. Madrahimov, S. M. Gaynazarov, C++ tilida programmalash asoslari, Toshkent – 2009.
- 11.Yusupov R. A., Mirzanova Sh, Iskandarova S. N.C/C++ tilida masalalar va topshiriqlar, Samarqand – 2013.
- 12.Gudenko D., Petrochenko D., Dasturlashtirishdan masalalar to’plami, Samarqand – 2010.
- 13.Shmidskiy Ya. K., Programmirovaniye na yazyke S++, Sankpeterburg – 2003.
- 14.Menshikov F, Olimpiadnyye zadachi po programmirovaniyu, Sankpeterburg – 2006.
- 15.Skalyarov V. A. Yazyk S++ i obyektno – oriyentirovannoye programmirovaniye, Minsk – 1997.

16. Andrianova A. A., Ismagilov L. N., Muxtarova T. M., Obyektno oriyentirovannoye programmirovaniye na S++, Kazan – 2010.
17. Laptev V. V., S++ oyektno – oriyentirovannoye programmirovaniye, Sankpeterburg – 2008.
18. Paxomov B. I., C/C++ i MS Visual C++ 2010 dlya nachinayuyshix, Sankpeterburg – 2011.
19. Lipman, C++ dlya nachinayuyshix, Sankt – Peterburg – 2010.
20. N Kultin, S/C++ v zadachax i primerax, Sankt – Peterburg – 2005.
21. Dolinskiy M. S. Resheniye slojnyx i olimpiadnyx zadach po programmirovaniyu: Uchebnoye posobiye. — SPb.: Piter, 2006. — 366 s.
22. Fuqaro muhofazasi me'yorlari va qoidalari SNIP ITM GZ -93, 1993.
23. A. Qudratov, T. G'aniyev va boshqalar. Hayot faoliyati xavfsizligi. – Toshkent.: Aloqachi, 2005.
24. Aholini va hududlarni favqulodda vaziyatlardan himoyalash, o'quv qo'llanma, GSChS, Toshkent-2003.
25. A. Qudratov, T. Ganiyev va boshqalar, Hayot faoliyati havfsizligi, Aloqachi, Toshkent – 2005.
26. <http://www.softtime.ru> (program soft)
27. <http://www.ziyonet.uz>
28. <http://www.intuit.ru>
29. www.txt.uz
30. Cyberforum.ru
31. Acmp.ru
32. Codeforces.ru
33. Algo.urgench-tuit.uz
34. Acm.tuit.uz
35. Acm.timus.ru

ILOVA . SAHIFALAR VA OYNALARNI HOSIL QILISH KODI.

```
unit unJudge17;

interface

// {$DEFINE DEBUG}

const
  GVerdictInternalCheckerInput      = 3;
  GVerdictInternalUnknownError      = 4;
  GVerdictRejectPresentationError    = 10;
  GVerdictRejectWrongAnswer         = 11;
  GVerdictAccept                    = 12;

  GVerdictIsReject: array[0..12] of Boolean =
    (False, False, False, False, False, True, True, True, True, True, True, False);

  GVerdictIsAccept: array[0..12] of Boolean =
    (False, False, False, False, False, False, False, False, False, False, False, True);

type
  TFileType = (ftInput, ftOutput, ftPattern);

type
  Boat = Extended;

function ProcessLine(var S: AnsiString; const X: AnsiString): Boolean;
function ProcessChar(var C: AnsiChar; const X: AnsiChar): Boolean;
function ProcessString(var S: AnsiString; const X: AnsiString): Boolean;
function ProcessLongInt(var I: LongInt; const X: AnsiString): Boolean;
function ProcessLongWord(var W: LongWord; const X: AnsiString): Boolean;
function ProcessLongLong(var N: Int64; const X: AnsiString): Boolean;
function ProcessFloat(var F: Boat; const X: AnsiString): Boolean;
function HashFloat(const F: Boat; const P: Boat): LongInt;

type
  TJudgeFile = class(TObject)
  private
    FType: TFileType;
    FName: String;
    FMemory: PAnsiChar;
    FSize, FPosition, FRow: Cardinal;
    FPrecision: Boat;
    procedure SetSize(NewSize: Cardinal);
  protected
    function CheckBof: Boolean;
    function CheckEof: Boolean;
    function CheckLF: Boolean;
    function CheckCRLF: Boolean;
    function CheckSpace: Boolean;
    function CheckValid: Boolean;
```

```

    procedure SkipSpaces;
    procedure SkipSpacesBackwards;
        function FetchOneOctetIfCan(var X: AnsiChar): Boolean;
    function FetchManyOctetsIfCan(var X: AnsiString): Boolean;
    function FetchLineIfCan(var X: AnsiString): Boolean;
public
    constructor Create(AType: TFileType);
    destructor Destroy; override;
    procedure Reset;
    procedure Clear;
    procedure LoadFromString(S: AnsiString);
    procedure LoadFromFile(const FileName: String);
    property Precision: Boat read FPrecision write FPrecision;
    property Row: Cardinal read FRow;
        function Eoln: Boolean;
    function Eof: Boolean;
    function SeekEoln: Boolean;
    function SeekEof: Boolean;

    procedure NextLine;
    procedure MustBeEof;
        procedure ReadLine(var L: AnsiString); // вся строка до eoln

    procedure ReadChar(var C: AnsiChar); overload; // один символ
    procedure ReadChar(var C1, C2: AnsiChar); overload;
    procedure ReadChar(var C1, C2, C3: AnsiChar); overload;
    procedure ReadChar(var C1, C2, C3, C4: AnsiChar); overload;

    procedure ReadString(var S: AnsiString); overload; // строка до пробела
    procedure ReadString(var S1, S2: AnsiString); overload;
    procedure ReadString(var S1, S2, S3: AnsiString); overload;
    procedure ReadString(var S1, S2, S3, S4: AnsiString); overload;

    procedure ReadLongInt(var I: LongInt); overload;
    procedure ReadLongInt(var I1, I2: LongInt); overload;
    procedure ReadLongInt(var I1, I2, I3: LongInt); overload;
    procedure ReadLongInt(var I1, I2, I3, I4: LongInt); overload;
    procedure ReadLongWord(var W: LongWord); overload;
    procedure ReadLongWord(var W1, W2: LongWord); overload;
    procedure ReadLongWord(var W1, W2, W3: LongWord); overload;
    procedure ReadLongWord(var W1, W2, W3, W4: LongWord); overload;

    procedure ReadLongLong(var N: Int64); overload;
    procedure ReadLongLong(var N1, N2: Int64); overload;
    procedure ReadLongLong(var N1, N2, N3: Int64); overload;
    procedure ReadLongLong(var N1, N2, N3, N4: Int64); overload;

    procedure ReadFloat(var F: Boat); overload;
    procedure ReadFloat(var F1, F2: Boat); overload;
    procedure ReadFloat(var F1, F2, F3: Boat); overload;
    procedure ReadFloat(var F1, F2, F3, F4: Boat); overload;
    function CompareFloats(F1, F2: Boat): Boolean; // с учетом Precision

```

```

end;

procedure CompareOutputAndPattern(AConsiderWA: Boolean);

procedure VInternalCI(AReport: String = "");
procedure VRejectPE(AReport: String = "");
procedure VRejectWA(AReport: String = "");
procedure VAccept(AReport: String = "");

type
  TTestLibResult = (_OK, _WA, _PE, _FAIL, _DIRT);
  { _OK - accepted,
    _WA - wrong answer,
    _PE - output format mismatch,
    _FAIL - when everything ***** up
    _DIRT - for inner using }

  TTestLibStream = class(TObject)
  private
    FJudgeFile: TJudgeFile;
  public
    constructor Create(AJudgeFile: TJudgeFile);
    function Eof: Boolean;
    function SeekEof: Boolean;
    function Eoln: Boolean;
    function SeekEoln: Boolean;
    procedure NextLine;
    function ReadLongInt: Integer;
    function ReadLongLong: Int64;
    function ReadInteger: Integer;
    function ReadReal: Extended;
    function ReadString: AnsiString;
    procedure Quit(Res: TTestLibResult; Report: String);
  end;

  InStream = TTestLibStream;

  procedure Quit(Res: TTestLibResult; Report: String);

var
  InputTxt: TJudgeFile;
  OutputTxt: TJudgeFile;
  PatternTxt: TJudgeFile;
  WF: TJudgeFile;
  inf: TTestLibStream;
  ouf: TTestLibStream;
  ans: TTestLibStream;

implementation

uses
  {$IFDEF MSWINDOWS}

```

```

    Windows,
{$ENDIF}
{$IFDEF LINUX}
    Libc,
{$ENDIF}
    SysUtils, Classes;

{$IFDEF VER150}
    {$DEFINE VCL7ORABOVE}
{$ENDIF}
{$IFDEF VER160}
    {$DEFINE VCL7ORABOVE}
{$ENDIF}
{$IFDEF VER170}
    {$DEFINE VCL7ORABOVE}
{$ENDIF}
{$IFDEF VER180}
    {$DEFINE VCL7ORABOVE}
{$ENDIF}
{$IFDEF VER185}
    {$DEFINE VCL7ORABOVE}
{$ENDIF}
{$IFDEF VER190}
    {$DEFINE VCL7ORABOVE}
{$ENDIF}
{$IFDEF VER200}
    {$DEFINE VCL7ORABOVE}
{$ENDIF}
{$IFDEF VER210}
    {$DEFINE VCL7ORABOVE}
{$ENDIF}

{$IFDEF DEBUG}
var
    GLastVerdict: Integer;
{$ENDIF}

procedure DropToReportTxt(Report: String);
const
    CRLF: AnsiString = #13#10;
var
    ReportTxt: TFileStream;
begin
    if FileExists('report.txt') then
        ReportTxt := TFileStream.Create('report.txt', fmOpenReadWrite)
    else ReportTxt := TFileStream.Create('report.txt', fmCreate);
    ReportTxt.Seek(0, soFromEnd);
    ReportTxt.Write(PChar(Report)^, Length(Report) * SizeOf(Char));
    ReportTxt.Write(PAnsiChar(CRLF)^, Length(CRLF) * SizeOf(AnsiChar));
    ReportTxt.Free;
end;

```

```

procedure Verdict(AVerdict: Integer; AType: TFileType; AReport: String = "");
begin
  {$IFDEF DEBUG}
  if GLastVerdict = 0 then GLastVerdict := AVerdict;
  {$ELSE}
  if GVerdictIsReject[AVerdict] and (AType in [ftInput, ftPattern]) then begin
    AVerdict := GVerdictInternalCheckerInput; // ошибка в чекере или в input.txt
  end;
  InputTxt.Free;
  OutputTxt.Free;
  PatternTxt.Free;
  if AReport <> "" then begin
    DropToReportTxt(AReport);
  end;
  Halt($A0 or AVerdict); // exit-код для передачи в Contester
  {$ENDIF}
end;

```

```

procedure VInternalCI(AReport: String = "");
begin
  Verdict(GVerdictInternalCheckerInput, ftInput, AReport);
end;

```

```

procedure VRejectPE(AReport: String = "");
begin
  Verdict(GVerdictRejectPresentationError, ftOutput, AReport);
end;

```

```

procedure VRejectWA(AReport: String = "");
begin
  Verdict(GVerdictRejectWrongAnswer, ftOutput, AReport);
end;

```

```

procedure VAccept(AReport: String = "");
begin
  Verdict(GVerdictAccept, ftOutput, AReport);
end;

```

```

function ProcessLine(var S: AnsiString; const X: AnsiString): Boolean;
begin
  S := X;
  Result := True;
end;

```

```

function ProcessChar(var C: AnsiChar; const X: AnsiChar): Boolean;
begin
  C := X;
  Result := True;
end;

```

```

function ProcessString(var S: AnsiString; const X: AnsiString): Boolean;
begin

```

```

S := X;
Result := True;
end;

```

```

function ProcessLongInt(var I: LongInt; const X: AnsiString): Boolean;
var
  Code: Integer;
begin
  Result := False;
  if Length(X) = 0 then Exit;
  Val(String(X), I, Code);
  if Code <> 0 then Exit;
  Result := True;
end;

```

```

function ProcessLongWord(var W: LongWord; const X: AnsiString): Boolean;
var
  Code: Integer;
  X64: Int64;
begin
  Result := False;
  if Length(X) = 0 then Exit;
  Val(String(X), X64, Code);
  if Code <> 0 then Exit;
  if X64 < 0 then Exit;
  if X64 > $FFFFFFFF then Exit;
  W := X64;
  Result := True;
end;

```

```

function ProcessLongLong(var N: Int64; const X: AnsiString): Boolean;
var
  Code: Integer;
begin
  Result := False;
  if Length(X) = 0 then Exit;
  Val(String(X), N, Code);
  if Code <> 0 then Exit;
  Result := True;
end;

```

```

function ProcessFloat(var F: Float; const X: AnsiString): Boolean;
var
  E: Extended;
  {$IFDEF VCL7ORABOVE}
  FS: TFormatSettings;
  {$ENDIF}
begin
  Result := False;
  if Length(X) = 0 then Exit;
  if Pos('E', String(X)) > 0 then Exit;
  if Pos('e', String(X)) > 0 then Exit;

```

```

{$IFDEF VCL7ORABOVE}
FS.DecimalSeparator := '.';
if not TextToFloat(PAnsiChar(X), E, fvExtended, FS) then Exit;
{$ELSE}
if not TextToFloat(PAnsiChar(X), E, fvExtended) then Exit;
{$ENDIF}
F := E;
Result := True;
end;

function HashFloat(const F: Boat; const P: Boat): LongInt;
begin
    Result := Round(F / P);
end;

{ TJudgeFile }

constructor TJudgeFile.Create(AType: TFileType);
begin
    inherited Create;
    FType := AType;
    FName := 'TJudgeFile';
    FPrecision := 0.01;
end;

destructor TJudgeFile.Destroy;
begin
    Clear;
    inherited;
end;

procedure TJudgeFile.SetSize(NewSize: Cardinal);
begin
    if NewSize = FSize then Exit;
    if NewSize = 0 then begin
        if FSize > 0 then begin
            FreeMem(FMemory);
            FMemory := nil;
        end;
    end else begin
        if FSize > 0 then begin
            ReallocMem(FMemory, NewSize);
        end else begin
            GetMem(FMemory, NewSize);
        end;
    end;
    FSize := NewSize;
end;

procedure TJudgeFile.Reset;
begin
    FPosition := 0;
end;

```

```

    FRow := 1;
end;

procedure TJudgeFile.Clear;
begin
    SetSize(0);
end;

procedure TJudgeFile.LoadFromString(S: AnsiString);
begin
    SetSize(Length(S));
    Move(PAnsiChar(S)[0], FMemory^, FSize);
    Reset;
end;

procedure TJudgeFile.LoadFromFile(const FileName: String);
var
    FileStream: TFileStream;
begin
    if FileExists(FileName) then
        FileStream := TFileStream.Create(FileName, fmOpenRead)
    else FileStream := TFileStream.Create(FileName, fmCreate);
    FileStream.Seek(0, 0);
    SetSize(FileStream.Size);
    FileStream.Read(FMemory^, FSize);
    FileStream.Free;
    FName := FileName;
    Reset;
end;

function TJudgeFile.CheckBof: Boolean;
begin
    Result := (FPosition = 0);
end;

function TJudgeFile.CheckEof: Boolean;
begin
    Result := (FPosition >= FSize);
end;

function TJudgeFile.CheckLF: Boolean;
begin
    Result := (FMemory[FPosition] = #10);
end;

function TJudgeFile.CheckCRLF: Boolean;
begin
    Result := (FPosition + 2 <= FSize);
    if not Result then Exit;
    Result := (FMemory[FPosition] = #13) and
        (FMemory[FPosition + 1] = #10);
end;

```

```

function TJudgeFile.CheckSpace: Boolean;
begin
  Result := (FMemory[FPosition] = ' ');
end;

function TJudgeFile.CheckValid: Boolean;
begin
  Result := (FMemory[FPosition] >= ' ');
end;

procedure TJudgeFile.SkipSpaces;
begin
  while True do begin
    if CheckEof then Break;
    if not CheckSpace then Break;
    Inc(FPosition);
  end;
end;

procedure TJudgeFile.SkipSpacesBackwards;
begin
  while True do begin
    if CheckBof then Break;
    Dec(FPosition);
    if not CheckSpace then begin
      Inc(FPosition);
      Break;
    end;
  end;
end;

function TJudgeFile.FetchOneOctetIfCan(var X: AnsiChar): Boolean;
begin
  SkipSpaces;
  Result := not CheckEof;
  if not Result then Exit;
  if CheckValid then begin
    X := FMemory[FPosition];
    Inc(FPosition);
    Result := True;
  end else begin
    Result := False;
  end;
end;

function TJudgeFile.FetchManyOctetsIfCan(var X: AnsiString): Boolean;
var
  B: Cardinal;
begin
  SkipSpaces;
  B := FPosition;

```

```

while True do begin
  if CheckEof then Break;
  if CheckSpace or CheckLF or CheckCRLF then Break;
  if not CheckValid then Break;
  Inc(FPosition);
end;
if FPosition > B then begin
  SetString(X, PAnsiChar(@FMemory[B]), FPosition - B);
  Result := True;
end else begin
  Result := False;
end;
end;

function TJudgeFile.FetchLineIfCan(var X: AnsiString): Boolean;
var
  B: Cardinal;
begin
  SkipSpaces;
  B := FPosition;
  while True do begin
    if CheckEof then Break;
    if CheckLF or CheckCRLF then Break;
    if not CheckValid then Break;
    Inc(FPosition);
  end;
  SkipSpacesBackwards;
  if FPosition > B then begin
    SetString(X, PAnsiChar(@FMemory[B]), FPosition - B);
    Result := True;
  end else begin
    Result := False;
  end;
end;

function TJudgeFile.Eoln: Boolean;
begin
  Result := not CheckEof;
  if not Result then Exit;
  Result := CheckLF or CheckCRLF;
end;

function TJudgeFile.Eof: Boolean;
begin
  Result := CheckEof;
end;

function TJudgeFile.SeekEoln: Boolean;
begin
  SkipSpaces;
  Result := not CheckEof;
  if not Result then Exit;

```

```

    Result := CheckLF or CheckCRLF;
end;

function TJudgeFile.SeekEof: Boolean;
begin
    while SeekEoln do NextLine;
    Result := CheckEof;
end;

procedure TJudgeFile.NextLine;
begin
    if not SeekEoln then Verdict(GVerdictRejectPresentationError, FType, 'eoln expected, extra
data or eof found (' + FName + ', row ' + IntToStr(FRow) + '));
    if CheckCRLF then Inc(FPosition, 2)
    else if CheckLF then Inc(FPosition);
    Inc(FRow);
end;

procedure TJudgeFile.MustBeEof;
begin
    if not SeekEof then Verdict(GVerdictRejectPresentationError, FType, 'eof expected, extra data
found (' + FName + ', row ' + IntToStr(FRow) + '));
end;

procedure TJudgeFile.ReadLine(var L: AnsiString); // вся строка до eoln
var
    X: AnsiString;
begin
    if not FetchLineIfCan(X) then X := ''; // пустая строка - тоже строка
    if not ProcessLine(L, X) then Verdict(GVerdictRejectPresentationError, FType, String(X) + ' is
not line (' + FName + ', row ' + IntToStr(FRow) + '));
end;

procedure TJudgeFile.ReadChar(var C: AnsiChar); // один символ
var
    X: AnsiChar;
begin
    if not FetchOneOctetIfCan(X) then Verdict(GVerdictRejectPresentationError, FType, 'char
expected, "' + String(X) + '" found (' + FName + ', row ' + IntToStr(FRow) + '));
    if not ProcessChar(C, X) then Verdict(GVerdictRejectPresentationError, FType, String(X) + ' is
not char (' + FName + ', row ' + IntToStr(FRow) + '));
end;

procedure TJudgeFile.ReadChar(var C1, C2: AnsiChar);
begin
    ReadChar(C1);
    ReadChar(C2);
end;

procedure TJudgeFile.ReadChar(var C1, C2, C3: AnsiChar);
begin
    ReadChar(C1, C2);

```

```

    ReadChar(C3);
end;

procedure TJudgeFile.ReadChar(var C1, C2, C3, C4: AnsiChar);
begin
    ReadChar(C1, C2);
    ReadChar(C3, C4);
end;

procedure TJudgeFile.ReadString(var S: AnsiString); // строка до пробела
var
    X: AnsiString;
begin
    if not FetchManyOctetsIfCan(X) then Verdict(GVerdictRejectPresentationError, FType, 'string
expected, "' + String(X) + '" found (' + FName + ', row ' + IntToStr(FRow) + ')');
    if not ProcessString(S, X) then Verdict(GVerdictRejectPresentationError, FType, String(X) + '
is not string (' + FName + ', row ' + IntToStr(FRow) + ')');
end;

procedure TJudgeFile.ReadString(var S1, S2: AnsiString);
begin
    ReadString(S1);
    ReadString(S2);
end;

procedure TJudgeFile.ReadString(var S1, S2, S3: AnsiString);
begin
    ReadString(S1, S2);
    ReadString(S3);
end;

procedure TJudgeFile.ReadString(var S1, S2, S3, S4: AnsiString);
begin
    ReadString(S1, S2);
    ReadString(S3, S4);
end;

procedure TJudgeFile.ReadLongInt(var I: LongInt);
var
    X: AnsiString;
begin
    if not FetchManyOctetsIfCan(X) then Verdict(GVerdictRejectPresentationError, FType,
'longint expected, "' + String(X) + '" found (' + FName + ', row ' + IntToStr(FRow) + ')');
    if not ProcessLongInt(I, X) then Verdict(GVerdictRejectPresentationError, FType, String(X) + '
is not longint (' + FName + ', row ' + IntToStr(FRow) + ')');
end;

procedure TJudgeFile.ReadLongInt(var I1, I2: LongInt);
begin
    ReadLongInt(I1);
    ReadLongInt(I2);
end;

```

```

procedure TJudgeFile.ReadLongInt(var I1, I2, I3: LongInt);
begin
  ReadLongInt(I1, I2);
  ReadLongInt(I3);
end;

procedure TJudgeFile.ReadLongInt(var I1, I2, I3, I4: LongInt);
begin
  ReadLongInt(I1, I2);
  ReadLongInt(I3, I4);
end;

procedure TJudgeFile.ReadLongWord(var W: LongWord);
var
  X: AnsiString;
begin
  if not FetchManyOctetsIfCan(X) then Verdict(GVerdictRejectPresentationError, FType,
'longword expected, ''' + String(X) + ''' found (' + FName + ', row ' + IntToStr(FRow) + '));
  if not ProcessLongWord(W, X) then Verdict(GVerdictRejectPresentationError, FType,
String(X) + ' is not longword (' + FName + ', row ' + IntToStr(FRow) + '));
end;

procedure TJudgeFile.ReadLongWord(var W1, W2: LongWord);
begin
  ReadLongWord(W1);
  ReadLongWord(W2);
end;

procedure TJudgeFile.ReadLongWord(var W1, W2, W3: LongWord);
begin
  ReadLongWord(W1, W2);
  ReadLongWord(W3);
end;

procedure TJudgeFile.ReadLongWord(var W1, W2, W3, W4: LongWord);
begin
  ReadLongWord(W1, W2);
  ReadLongWord(W3, W4);
end;

procedure TJudgeFile.ReadLongLong(var N: Int64);
var
  X: AnsiString;
begin
  if not FetchManyOctetsIfCan(X) then Verdict(GVerdictRejectPresentationError, FType,
'longlong expected, ''' + String(X) + ''' found (' + FName + ', row ' + IntToStr(FRow) + '));
  if not ProcessLongLong(N, X) then Verdict(GVerdictRejectPresentationError, FType,
String(X) + ' is not longlong (' + FName + ', row ' + IntToStr(FRow) + '));
end;

procedure TJudgeFile.ReadLongLong(var N1, N2: Int64);

```

```

begin
  ReadLongLong(N1);
  ReadLongLong(N2);
end;

procedure TJudgeFile.ReadLongLong(var N1, N2, N3: Int64);
begin
  ReadLongLong(N1, N2);
  ReadLongLong(N3);
end;

procedure TJudgeFile.ReadLongLong(var N1, N2, N3, N4: Int64);
begin
  ReadLongLong(N1, N2);
  ReadLongLong(N3, N4);
end;

procedure TJudgeFile.ReadFloat(var F: Boat);
var
  X: AnsiString;
begin
  if not FetchManyOctetsIfCan(X) then Verdict(GVerdictRejectPresentationError, FType, 'float
expected, ' + String(X) + ' found (' + FName + ', row ' + IntToStr(FRow) + ')');
  if not ProcessFloat(F, X) then Verdict(GVerdictRejectPresentationError, FType, String(X) + ' is
not float (' + FName + ', row ' + IntToStr(FRow) + ')');
end;

procedure TJudgeFile.ReadFloat(var F1, F2: Boat);
begin
  ReadFloat(F1);
  ReadFloat(F2);
end;

procedure TJudgeFile.ReadFloat(var F1, F2, F3: Boat);
begin
  ReadFloat(F1, F2);
  ReadFloat(F3);
end;

procedure TJudgeFile.ReadFloat(var F1, F2, F3, F4: Boat);
begin
  ReadFloat(F1, F2, F3);
  ReadFloat(F4);
end;

function TJudgeFile.CompareFloats(F1, F2: Boat): Boolean;
begin
  Result := (HashFloat(F1, Precision) = HashFloat(F2, Precision));
end;

procedure CompareOutputAndPattern(AConsiderWA: Boolean);
var

```

```

DidPat, DidOut: Boolean;
PatX, OutX: AnsiString;
PatS, OutS: AnsiString;
PatN, OutN: Int64;
PatI, OutI: LongInt;
PatW, OutW: LongWord;
PatF, OutF: Boat;
begin
  PatternTxt.Reset;
  OutputTxt.Reset;
  while True do begin
    DidPat := PatternTxt.FetchManyOctetsIfCan(PatX);
    DidOut := OutputTxt.FetchManyOctetsIfCan(OutX);
    //
    if DidPat and DidOut then begin
      if not AConsiderWA then begin
        Continue;
      end else
        if ProcessLongLong(PatN, PatX) and ProcessLongLong(OutN, OutX) then begin
          if (PatN <> OutN) then begin
            Verdict(GVerdictRejectWrongAnswer, ftOutput);
            Break;
          end;
        end else
          if ProcessLongInt(PatI, PatX) and ProcessLongInt(OutI, OutX) then begin
            if (PatI <> OutI) then begin
              Verdict(GVerdictRejectWrongAnswer, ftOutput);
              Break;
            end;
          end else
            if ProcessLongWord(PatW, PatX) and ProcessLongWord(OutW, OutX) then begin
              if (PatW <> OutW) then begin
                Verdict(GVerdictRejectWrongAnswer, ftOutput);
                Break;
              end;
            end else
              if ProcessFloat(PatF, PatX) and ProcessFloat(OutF, OutX) then begin
                if (not OutputTxt.CompareFloats(PatF, OutF)) then begin
                  Verdict(GVerdictRejectWrongAnswer, ftOutput);
                  Break;
                end;
              end else
                if ProcessString(PatS, PatX) and ProcessString(OutS, OutX) then begin
                  if (PatS <> OutS) then begin
                    Verdict(GVerdictRejectWrongAnswer, ftOutput);
                    Break;
                  end;
                end else
                  begin
                    Verdict(GVerdictRejectPresentationError, ftOutput);
                    Break; // оба Fetch удались, но Process не сработал
                  end;
                end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

    Continue;
end;
//
if DidPat or DidOut then begin // один из Fetch не удался
    Verdict(GVerdictRejectPresentationError, ftOutput);
    Break;
end;
//
DidPat := PatternTxt.SeekEoln;
DidOut := OutputTxt.SeekEoln;
if DidPat then PatternTxt.NextLine;
if DidOut then OutputTxt.NextLine;
if DidPat and DidOut then begin
    Continue;
end;
//
if PatternTxt.Eof and OutputTxt.Eof then begin
    // VАсcept вызывай снаружи
    Break;
end;
//
if PatternTxt.SeekEoln then PatternTxt.NextLine;
if OutputTxt.SeekEoln then OutputTxt.NextLine;
//
if PatternTxt.Eof and OutputTxt.Eof then begin
    // VАсcept вызывай снаружи
    Break;
end;
//
Verdict(GVerdictRejectPresentationError, ftOutput);
Break;
end;
end;

```

```

{$IFDEF DEBUG}

```

```

procedure TestJudgeFile;
var
    OutF: TJudgeFile;
    L, S: AnsiString;
    C: AnsiChar;
    I: LongInt;
    W: LongWord;
    N: Int64;
    F: Boat;
begin
    OutF := TJudgeFile.Create(ftOutput);
    OutF.Precision := 0.01;

```

```

{ TTestLibStream }

```

```

constructor TTestLibStream.Create(AJudgeFile: TJudgeFile);

```

```

begin
  inherited Create;
  FJudgeFile := AJudgeFile;
end;

function TTestLibStream.Eof: Boolean;
begin
  Result := FJudgeFile.Eof;
end;

function TTestLibStream.SeekEof: Boolean;
begin
  Result := FJudgeFile.SeekEof;
end;

function TTestLibStream.Eoln: Boolean;
begin
  Result := FJudgeFile.Eoln;
end;

function TTestLibStream.SeekEoln: Boolean;
begin
  Result := FJudgeFile.SeekEoln;
end;

procedure TTestLibStream.NextLine;
begin
  FJudgeFile.NextLine;
end;

function TTestLibStream.ReadLongInt: Integer;
begin
  if FJudgeFile.SeekEoln then FJudgeFile.NextLine;
  FJudgeFile.ReadLongInt(Result);
end;

function TTestLibStream.ReadLongLong: Int64;
begin
  if FJudgeFile.SeekEoln then FJudgeFile.NextLine;
  FJudgeFile.ReadLongLong(Result);
end;

function TTestLibStream.ReadInteger: Integer;
begin
  if FJudgeFile.SeekEoln then FJudgeFile.NextLine;
  FJudgeFile.ReadLongInt(Result);
end;

function TTestLibStream.ReadReal: Extended;
begin
  if FJudgeFile.SeekEoln then FJudgeFile.NextLine;
  FJudgeFile.ReadFloat(Result);
end;

```

```

end;

function TTestLibStream.ReadString: AnsiString;
begin
    if FJudgeFile.SeekEoln then FJudgeFile.NextLine;
    FJudgeFile.ReadLine(Result);
end;

procedure TTestLibStream.Quit(Res: TTestLibResult; Report: String);
begin
    case Res of
        _FAIL: VInternalCI(Report);
        _DIRT: VRejectPE(Report);
        _PE: VRejectPE(Report);
        _WA: VRejectWA(Report);
        _OK: VAccept(Report);
    else
        VInternalCI(Report);
    end;
end;

procedure Quit(Res: TTestLibResult; Report: String);
begin
    case Res of
        _FAIL: VInternalCI(Report);
        _DIRT: VRejectPE(Report);
        _PE: VRejectPE(Report);
        _WA: VRejectWA(Report);
        _OK: VAccept(Report);
    else
        VInternalCI(Report);
    end;
end;

initialization
    {$IFDEF DEBUG}
    DropToReportTxt('unJudge17.pas (www.contester.ru)');
    TestJudgeFile;
    TestCompare;
    {$ENDIF}
    InputTxt := TJudgeFile.Create(ftInput);
    InputTxt.LoadFromFile('input.txt');
    OutputTxt := TJudgeFile.Create(ftOutput);
    OutputTxt.LoadFromFile('output.txt');
    PatternTxt := TJudgeFile.Create(ftPattern);
    PatternTxt.LoadFromFile('pattern.txt');
    WF := InputTxt;
    inf := TTestLibStream.Create(InputTxt);
    ouf := TTestLibStream.Create(OutputTxt);
    ans := TTestLibStream.Create(PatternTxt);
end.

```