MTA Kredit4-labaratoriya ishi

DINAMIK MA’LUMOTLAR TUZILMASINI TADQIQ QILISH. CHIZIQLI BOG’LANGAN RO’YHATLAR.

Ishdan maqsad: Chiziqli, bir bog‘lamli ro‘yhatlar tuzilmasini o‘rganish va uni ustida amal bajarish algoritmlarini tadqiq qilish.

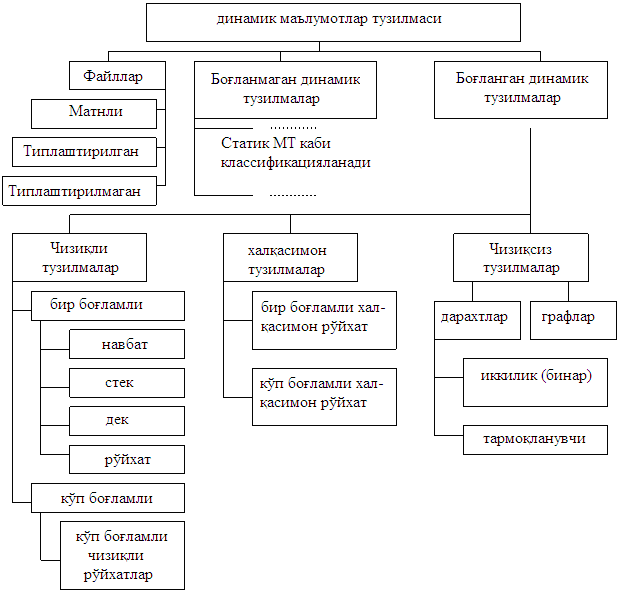
Qo‘yilgan masala: C++ tilida ro‘yhatli tuzilma elementlarini ko‘rsatkichli maydonlar bilan yaratish va dinamik tuzilmani e’lon qilish, uning ustida turli amallar bajarish dasturini ishlab chiqish.

Ish tartibi:

* Tajriba ishi nazariy ma’lumotlarini o‘rganish;
* Berilgan topshiriqning algoritmini ishlab chiqish;
* C++ dasturlash muhitida dasturni yaratish;
* Natijalarni tekshirish;
* Hisobotni tayyorlash va topshirish.

## 3.1. Dinamik ma’lumotlar tuzilmasi

Statik ma’lumotlar tuzilmasi vaqt o‘tishi bilan o‘z o‘lchamini o‘zgartirmaydi. Biz har doim dastur kodidagi statik ma’lumotlar tuzilmasiga qarab ularning o‘lchamini bilishimiz mumkin. Bunday ma’lumotlarga teskari ravishda dinamik ma’lumotlar tuzilmasi mavjud bo‘lib, bunda dastur bajarilishi davomida dinamik ma’lumotlar tuzilmasi o‘lchamini o‘zgartirishi mumkin***.*** *Dinamik ma’lumotlar tuzilmasi* – bu qandaydir bir qonuniyatga asoslanib shakllangan, lekin elementlari soni, o‘zaro joylashuvi va o‘zaro aloqasi dastur bajarilishi davomida shu qonuniyat asosida dinamik o‘zgaruvchan bo‘lgan ma’lumotlar tuzilmasidir. Dinamik ma’lumotlar tuzilmasi 3.1-rasmdagidek klassifikatsiyalanadi.



Dinamik ma’lumotlat tuzilmasi

fayllar

matnli

toifalashtirilgan

toifalashtirilmagan

Bog‘lanmagan dinamik tuzilmalar

Statik MT kabi klassifikasiyalanadi

Bog‘langan dinamik tuzilmalar

chiziqli tuzilmalar

Halqasimon tuzilmalar

chiziqsiz tuzilmalar

Bir bog‘lamli

navbat

stek

dek

ro‘yhat

ko‘p bog‘lamli

bir bog‘lamli hal- qasimon ro‘yhatlar

ko‘p bog‘lamli hal- qasimon ro‘yhatlar

daraxtlar

graflar

Ikkilik (binar)

tarmoqlanuvchi

ko‘p bog‘lamli chiziqli ro‘yhatlar

3.1-rasm. Dinamik ma’lumotlar tuzilmasi klassifikatsiyasi

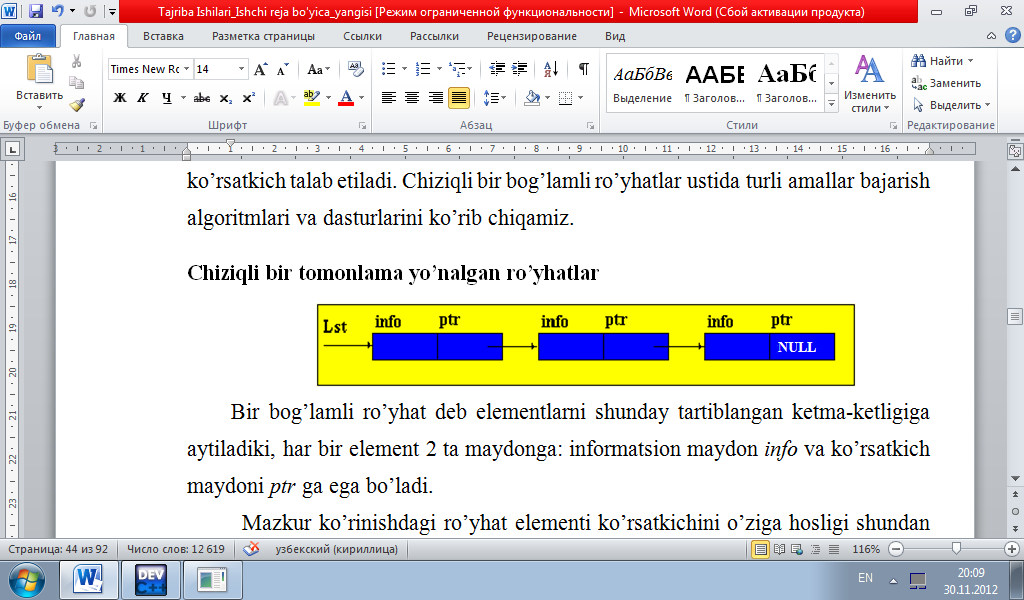
Dasturlarda dinamik ma’lumotlar tuzilmasidan ko‘pincha chiziqli ro‘yhatlar, steklar, navbatlar va binar daraxtlar ishlatiladi. Bu tuzilmalar bir-biridan elementlarning bog‘lanish usuli va ular ustida bajarilishi mumkin bo‘lgan amallari bilan farqlanadi. Dinamik tuzilmalar massiv va yozuvdan farqli ravishda operativ xotirada ketma-ket sohalarda joylashmaydi. Ixtiyoriy dinamik tuzilma elementi 2 ta maydondan tashkil topadi: tuzilma tashkil etilishiga sabab bo‘layotgan

*informatsion maydon* va elementlarning o‘zaro aloqasini ta’minlovchi *ko‘rsatkichli maydon*. Chiziqli ro‘yhatlarda har bir element o‘zidan keyingisi yoki oldingisi bilan ham bog‘langan bo‘lishi mumkin. Birinchi holatda, ya’ni elementlar o‘zidan keyingi element bilan bog‘langan bo‘lsa, bunday ro‘yhatga *bir bog‘lamli ro‘yhat* deyiladi. Agar har bir element o‘zidan oldingi va o‘zidan keyingi element bilan bog‘langan bo‘lsa, u holda bunday ro‘yhatlarga *2 bog‘lamli ro‘yhatlar* deyiladi. Agar oxirgi element birinchi element ko‘rsatkichi bilan bog‘langan bo‘lsa, bunday ro‘yhatga *halqasimon ro‘yhat* deyiladi. Ro‘yhatning har bir elementi shu elementni identifikatsiyalash uchun *kalit*ga ega bo‘ladi. Kalit odatda butun son yoki satr ko‘rinishida ma’lumotlar maydonining bir qismi sifatida mavjud bo‘ladi. Ro‘yhatlar ustida quyidagi amallarni bajarish mumkin.

* ro‘yhatni shakllantirish (birinchi elementini yaratish);
* ro‘yhat oxiriga yangi element qo‘shish;
* berilgan kalitga mos elementni o‘qish;
* ro‘yhatning ko‘rsatilgan joyiga element qo‘shish (berilgan kalitga mos elementdan oldin yoki keyin)
* berilgan kalitga mos elementni o‘chirish;
* kalit bo‘yicha ro‘yhat elementlarini tartibga keltirish.

Ro‘yhatlar bilan ishlashda dasturda boshlang‘ich elementni ko‘rsatuvchi ko‘rsatkich talab etiladi. Chiziqli bir bog‘lamli ro‘yhatlar ustida turli amallar bajarish algoritmlari va dasturlarini ko‘rib chiqamiz.

### 3.2. Chiziqli bir tomonlama yo‘nalgan ro‘yhatlar



3.2-rasm. Chiziqli bir bog‘lamli ro‘yhatlar

Bir bog‘lamli ro‘yhat deb elementlarning shunday tartiblangan ketma-ketligiga aytiladiki, har bir element 2 ta maydonga: informatsion maydon *info* va ko‘rsatkich maydoni *ptr* ga ega bo‘ladi (3.2-rasm).

Mazkur ko‘rinishdagi ro‘yhat elementi ko‘rsatkichining o‘ziga xosligi shundan iboratki, u faqatgina ro‘yhatning navbatdagi (o‘zidan keyin keluvchi) elementi adresini ko‘rsatadi. Bir tomonlama yo‘naltirilgan ro‘yhatda eng so‘ngi element ko‘rsatkichi bo‘sh, ya’ni *NULL* bo‘ladi.

*Lst* – ro‘yhat boshi ko‘rsatkichi. U ro‘yhatni yagona bir butun sifatida ifodalaydi. Ba’zan ro‘yhat bo‘sh bo‘lishi ham mumkin, ya’ni ro‘yhatda bitta ham element bo‘lmasligi mumkin. Bu holda *lst = NULL* bo‘ladi. Hozir chiziqli bir bog‘lamli ro‘yhat hosil qilish dasturini ko‘rib chiqsak. Buning uchun biz foydalanuvchi tomonidan yaratiladigan nostandart toifa yaratib olishimiz kerak. Buning bir qancha usullari mavjud, ya’ni klasslar yoki strukturalar bilan amalga oshirish mumkin. Masalan,

*class Node{*

*public://klass ma’lumotlariga tashqaridan bo‘ladigan murojaatga ruxsat berish*

*int info; // informatsion maydon*

*Node\* next;// ko‘rsatkichli maydon*

*};*

Bu yerda biz *Node*  nomli toifa yaratdik va ro‘yhatimiz butun sonlardan iborat. Endi ro‘yhat elementlarini shu toifa orqali e’lon qilsak bo‘ladi, ya’ni

*Node \*lst = NULL;// ro‘yhat boshi ko‘rsatkichi*

*Node \*last = NULL;// ro‘yhatga oxirgi kelib tushgan elementning ko‘rsatkichi*

Endi shu belgilashlar orqali ro‘yhat hosil qilamiz.

*Node \* p = new Node;*

*int numb = -1;*

*cout<<"son kiriting: ";*

*cin>>numb;*

*p->info = numb;*

*p->next = NULL;*

*if (lst == NULL) {*

*lst = p;*

*last = p;*

*}*

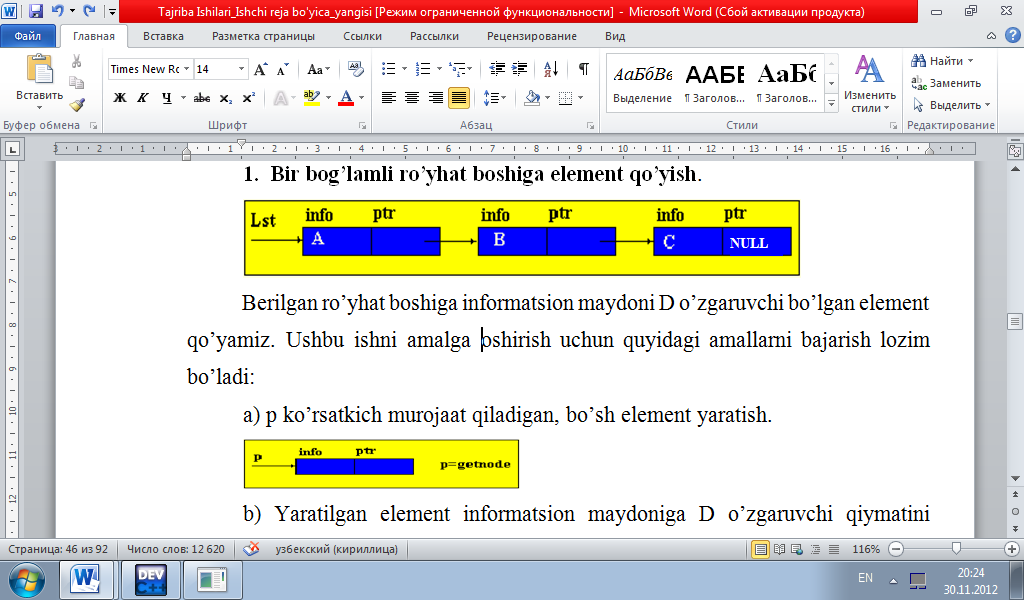
*else{ last->next = p;*

*last = p; }*

Bu dasturda yangi element ro‘yhat oxiridan kelib qo‘shiladi.

Bir bog‘lamli ro‘yhatlar ustida amallar bajarish algoritmlari

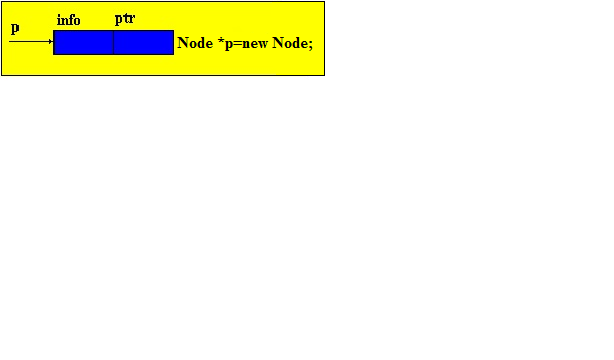
1. **Bir bog‘lamli ro‘yhat boshiga element qo‘yish**



3.3-rasm. Bir bog‘lamli chiziqli ro‘yhat tuzilishi

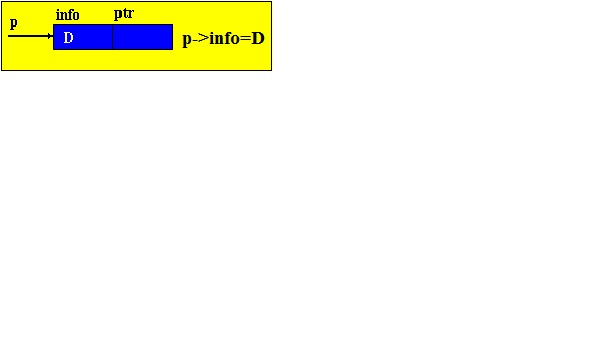
3.3-rasmdagi ro‘yhat boshiga informatsion maydoni D o‘zgaruvchi bo‘lgan element qo‘yamiz. Ushbu ishni amalga oshirish uchun quyidagi amallarni bajarish lozim bo‘ladi:

a) pko‘rsatkich murojaat qiladigan, bo‘sh element yaratish (3.4-rasm).



3.4-rasm. Yangi element hosil qilish

b) Yaratilgan element informatsion maydoniga D o‘zgaruvchi qiymatini o‘zlashtirish (3.5-rasm).

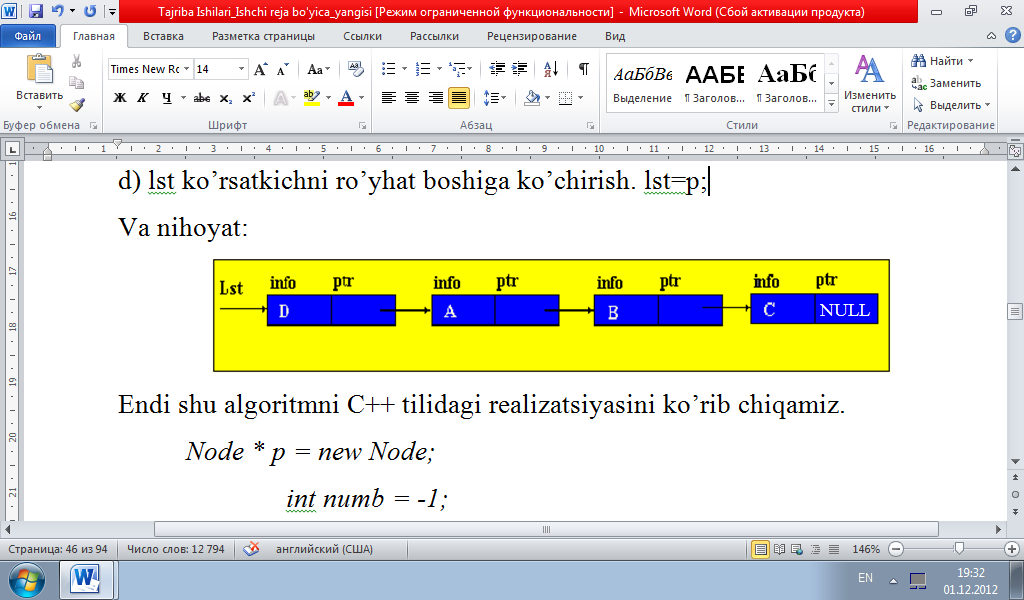


3.5-rasm. Yangi element info maydoniga qiymat kiritish

c) Yangi elementni ro‘yhat bilan bog‘lash: p->ptr=lst; (shu holatda yangi element va lst – ro‘yhat boshini ko‘rsatyapti)

d) lst ko‘rsatkichni ro‘yhat boshiga ko‘chirish (3.6-rasm). lst=p;

Va nihoyat:



3.6-rasm. Ro‘yhat boshiga element qo‘shish

Endi shu algoritmni C++ tilidagi realizatsiyasini ko‘rib chiqamiz.

*Node \* p = new Node;*

*int numb = -1;*

*cout<<"son kiriting: ";*

*cin>>numb;*

*p->info = numb;*

*if (lst ==NULL){*

*p->next = NULL;*

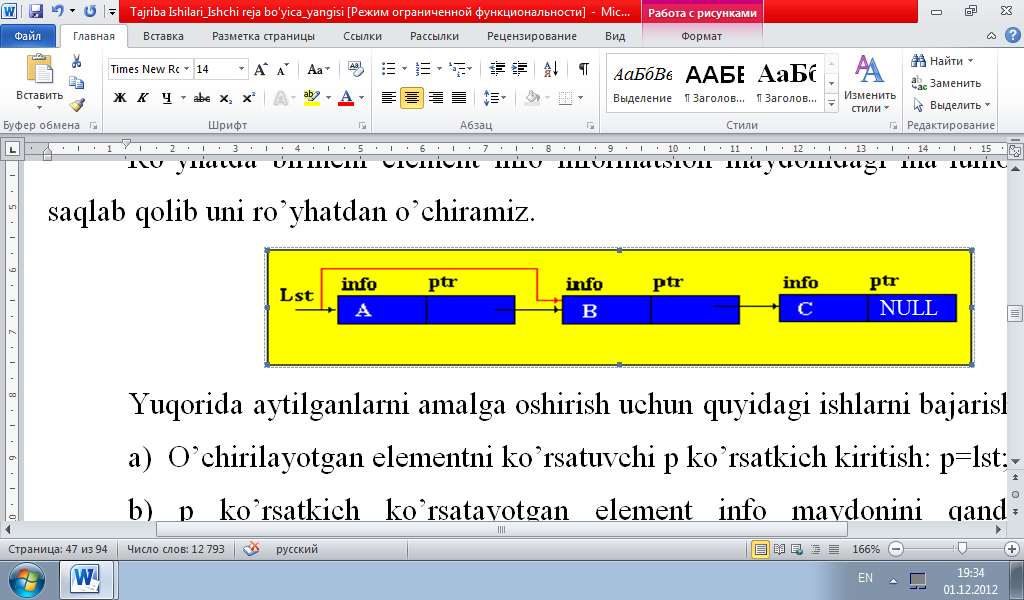
*lst = p; }*

*else { p->next = lst;*

*lst = p;}*

### 2. Bir bog‘lamli ro‘yhat boshidan elementni o‘chirish

Ro‘yhatda birinchi element info informatsion maydonidagi ma’lumotni esda saqlab qolib uni ro‘yhatdan o‘chiramiz (3.7-rasm).



3.7-rasm. Ro‘yhat boshidagi elementni o‘chirish

Yuqorida aytilganlarni amalga oshirish uchun quyidagi ishlarni bajarish lozim:

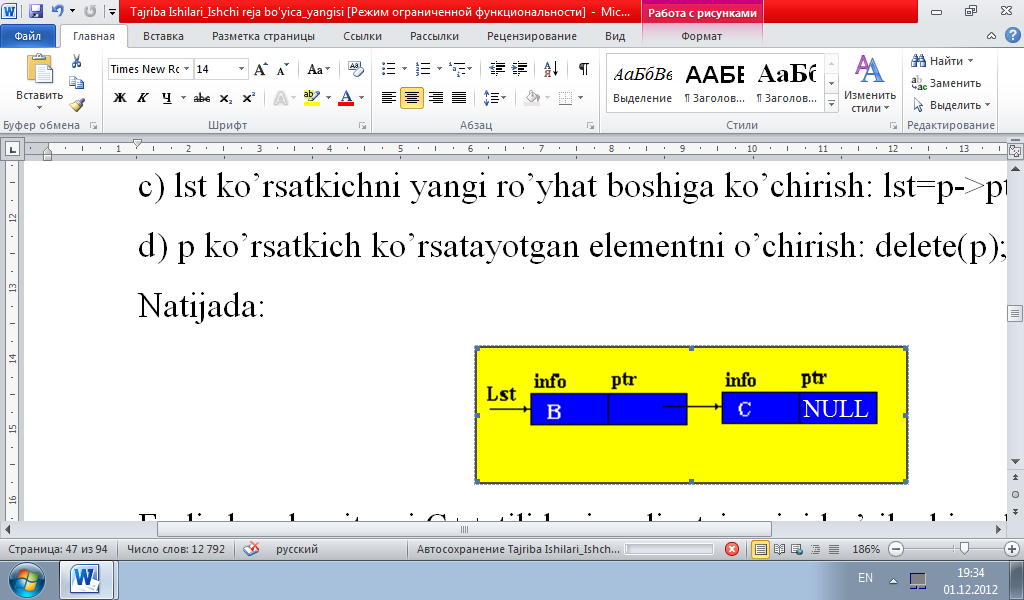
a) o‘chirilayotgan elementni ko‘rsatuvchi p ko‘rsatkich kiritish: p=lst;

b) p ko‘rsatkich ko‘rsatayotgan element info maydonini qandaydir x o‘zgaruvchida saqlash: x=p->info;

c) lst ko‘rsatkichni yangi ro‘yhat boshiga ko‘chirish: lst=p->ptr;

d) p ko‘rsatkich ko‘rsatayotgan elementni o‘chirish: delete(p);

Natijada 3.8-rasmdagi ko‘rinishga ega bo‘lamiz.



3.8-rasm. Ro‘yhatning natijaviy ko‘rinishi

Endi shu algoritmni C++ tilidagi realizatsiyasini ko‘rib chiqsak.

*Node\* p = new Node;*

*if (lst == NULL){*

*cout<<"ro'yhat bo'sh";*

*system("pause");*

*system("CLS");*

*}*

*else { p = lst;*

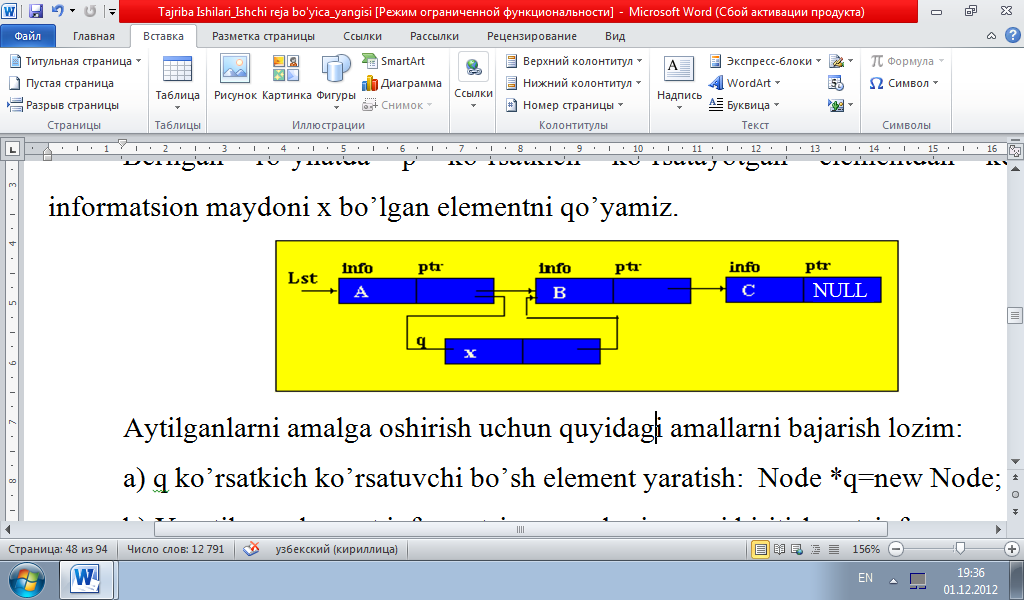
*lst = p->next ;*

*delete(p);*

*}*

### 3. Elementni ro‘yhatga qo‘shish

Berilgan ro‘yhatda p ko‘rsatkich ko‘rsatayotgan elementdan keyin informatsion maydoni x bo‘lgan elementni qo‘yamiz (3.9-rasm).



3.9-rasm. Ro‘yhatga yangi element qo‘shish

Aytilganlarni amalga oshirish uchun quyidagi amallarni bajarish lozim:

a) q ko‘rsatkich ko‘rsatuvchi bo‘sh elementni yaratish: Node \*q=new Node;

b) Yaratilgan element informatsion maydoniga x ni kiritish: q->info=x;

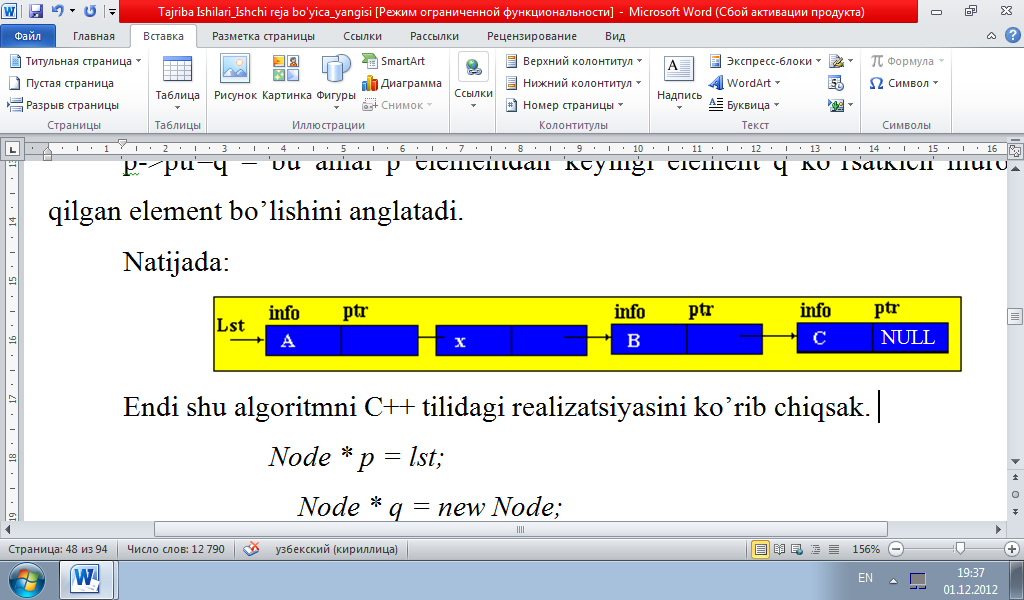
c) q elementni p elementdan keyingi element bilan bog‘lash.

q->ptr=p->ptr – yaratilgan element ko‘rsatkichiga p element ko‘rsatkichini o‘zlashtirish.

d) p element bilan q elementni bog‘lash.

p->ptr=q – bu amal p elementdan keyingi element q ko‘rsatkich murojaat qilgan element bo‘lishini anglatadi.

Natijada quyidagi rasmdagidek ko‘rinishga ega bo‘lamiz.



3.10-rasm. Natijaviy ro‘yhat ko‘rinishi

Endi shu algoritmni C++ tilidagi realizatsiyasini ko‘rib chiqsak.

*Node \* p = lst;*

*Node \* q = new Node;*

*int numb = -1;*

*cout<<"son kiriting: ";*

*cin>>numb;*

*q->number = numb;*

*int k;*

*cout<<"nechta elementdan keyin kiritasiz k=";cin>>k;*

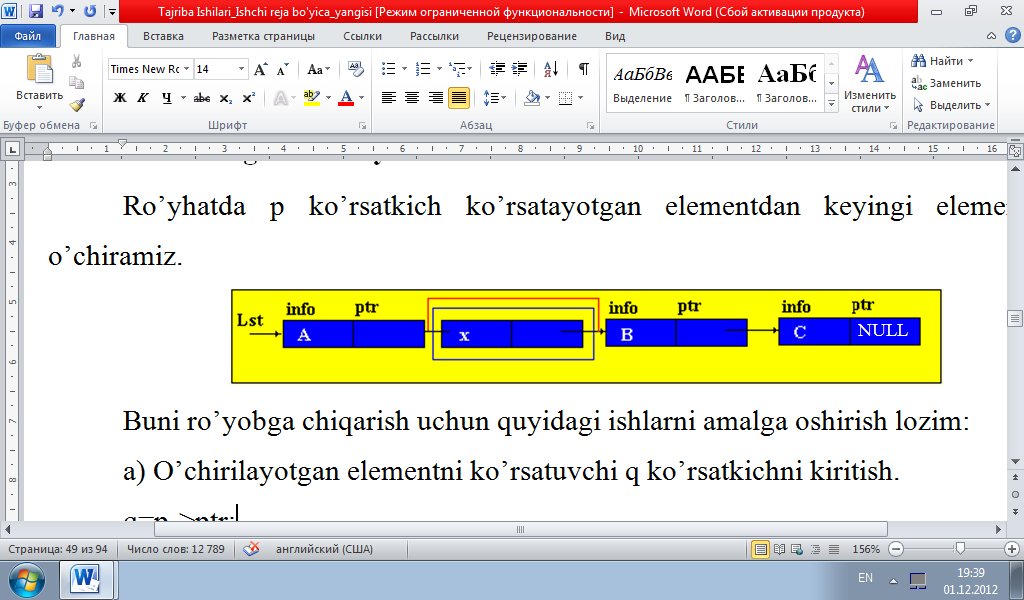
*for(int i=0;i<k-1;i++) p=p->next;*

*q->next = p->next;*

*p->next = q;*

### 4. Bir bog‘lamli ro‘yhatdan elementni o‘chirish

Ro‘yhatda p ko‘rsatkich ko‘rsatayotgan elementdan keyingi elementni o‘chiramiz (3.11-rasm).



3.11-rasm. Ro‘yhat o‘rtasidan element o‘chirish

Buni ro‘yobga chiqarish uchun quyidagi ishlarni amalga oshirish lozim:

a) O‘chirilayotgan elementni ko‘rsatuvchi q ko‘rsatkichni kiritish.

q=p->ptr;

b) p elementni q elementdan keyingi element bilan bog‘lash.

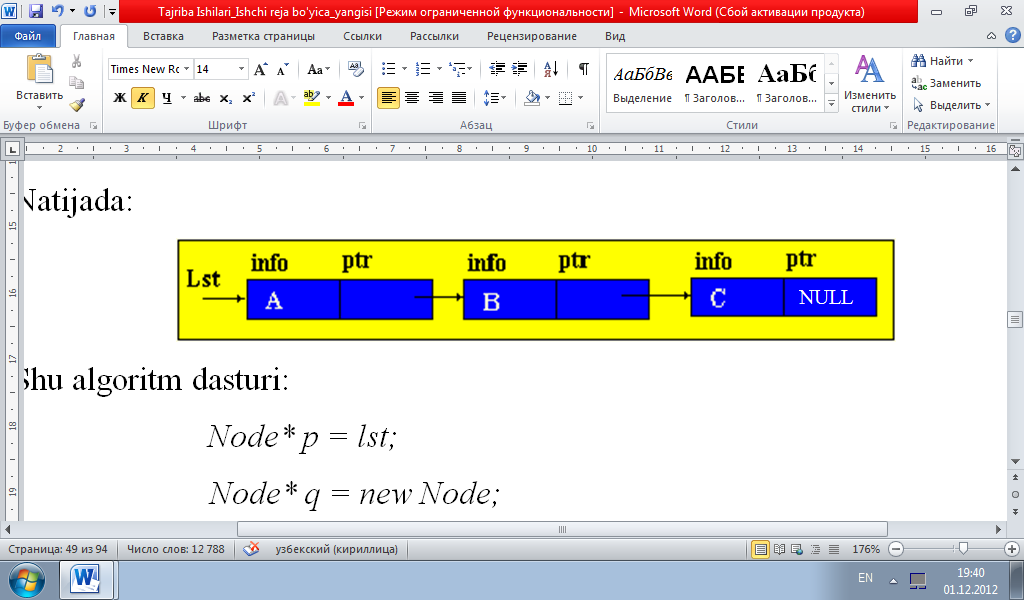
p->ptr=q->ptr;

c) O‘chirilayotgan element info maydonidagi informatsiyani yodda saqlash (agar zarur bo‘lsa) k=q->info;

d) q ko‘rsatkich ko‘rsatayotgan elementni o‘chirish.

delete(q)

Natijada ro‘yhat quyidagi ko‘rinishga ega bo‘ladi:



3.12-rasm. Natijaviy ro‘yhat ko‘rinishi

Shu algoritm dasturi:

*Node\* p = lst;*

*Node\* q = new Node;*

*int k;*

*cout<<"k=";cin>>k;*

*for(int i=0;i<k-1;i++) p=p->next;*

*q = p->next;*

*p->next = q->next;*

*delete(q);*

**Ishni bajarishga namuna**

Topshiriq variantlariga o’xshash bitta misolni yechish dasturini ko’rib chiqamiz. Quyidagicha masala qo’yilgan bo’lsin. Ro’yhatning maksimal elementi topilsin. Ushbu masalaning algoritmi, dasturiy kodi va natijasi quyida keltirilgan.

**Algoritm**

1. Ekranga menyu chiqaramiz: 1 - *element qo’shish; 2 - ro’yhatni ko’rish; 3 - ro’yhat maksimalini topish; 0 - chiqish;* tanlash uchun *tanla* o’zgaruvchisiga qiymat so’raymiz. 2-qadamga o’tish.
2. Agar *tanla*=1 bo’lsa, 3-qadamga, 2 ga teng bo’lsa, 4-qadamga, 3 tanlansa, 6-qadamga o’tish, 0 tanlansa dasturni yakunlash.
3. Navbatdagi elementni yaratish p; (p ning info maydoniga qiymat so’rab olib yozish va ptr maydoniga NULL yozish) Agar ro’yhat boshi ko’rsatkichi lst=NULL bo’lsa, lst=p va last=p; aks holda last – ro’yhat oxirgi elementi ptr maydoniga p ni yozib, p elementni last qilib belgilaymiz. 1-qadamga o’tamiz.
4. Agar lst NULL ga teng bo’lsa, ro’yhat bo’shligini ekranga chiqarib, 1-qadamga o’tish. Aks holda, p=lst va 5-qadamga o’tish.
5. Agar p ning ptr maydoni NULL bo’lmasa, p ning info maydonini ekranga chiqaramiz va keyingi elementga o’tamiz, ya’ni p=p->ptr, 5-qadamga o’tamiz, aks holda, 1-qadamga o’tamiz.
6. max=lst->info, ya’ni max o’zgaruvchisiga ro’yhat 1-elementi info maydoni qiymatini o’zlashtiramiz. p=lst va 7-qadamga o’tish.
7. Agar p NULL ga teng bo’lmasa, 8-qadamga o’tamiz, aks holda max ni ekranga chiqaramiz va 1-qadamga o’tamiz.
8. Agar max< p->info bo’lsa, max=p->info. Keyingi elementga o’tamiz, ya’ni p=p->ptr. 7-qadamga o’tamiz.

**Dastur kodi**

*#include <iostream>*

*using namespace std;*

*class Node{*

*public: int number;*

*Node\* next;*

*};*

*int main()*

*{ Node\* head = NULL;*

*Node\* lastPtr = NULL;*

*short action = -1;*

*while (1)*

*{ cout<<"1. element qo’shish\n";*

*cout<<"2. ro’yhatni ko’rish\n";*

*cout<<"3. ro’yhat maksimalini topish\n";*

*cout<<"0. chiqish\n\n";*

*cout<<"tanlang: ";*

*cin>>action;*

*if (action == 0) {*

*system("CLS");*

*break;}*

*if (action == 1)*

*{ system("CLS");*

*Node\* ptr = new Node;*

*int numb = -1;*

*cout<<"son kiriting: ";*

*cin>>numb;*

*ptr->number = numb;*

*ptr->next = NULL;*

*if (head == 0)*

*{ head = ptr;*

*lastPtr = ptr;*

*system("CLS");*

*continue;*

*}*

*lastPtr->next = ptr;*

*lastPtr = ptr;*

*system("CLS");*

*continue;*

*}*

*if (action == 2){*

*Node\* ptr = NULL;*

*system("CLS");*

*if (head == 0)*

*{ cout<<"\t!!! ro’yhat bo’sh !!!\n\n";*

*system("PAUSE");*

*system("CLS");*

*continue;*

*}*

*cout<<"\* \* \* \* \* ro’yhat \* \* \* \* \*\n\n";*

*ptr = head;*

*while (1) {*

*cout<<ptr->number<<" ";*

*if (ptr->next == 0) break;*

*ptr = ptr->next;*

*}*

*cout<<"\n\n";*

*system("PAUSE");*

*system("CLS");*

*continue;*

*}*

*if (action == 3)*

*{*

*system("CLS");*

*Node\* p = head;*

*Node\* q = new Node;*

*Node\* last = new Node;*

*int max=p->number; q=head;*

*while(p){*

*if(max<p->number){ max=p->number;}*

*p=p->next;*

*}*

*system("CLS");*

*cout<<"max="<<max;*

*system("pause");*

*continue;*

*}*

*}}*

**Dastur bajarilishi natijasi**

1. *element qo’shish*
2. *ro’yhatni ko’rish*
3. *ro’yhat maksimalini topish*
4. *chiqish*

*tanlang:1*

{1,2,3,55,4,6} sonlari kiritildi. 2-holat tanlanganda natija:

*\*\*\*\*\*ro’yhat\*\*\*\*\**

*1 2 3 55 4 6*

3-holat tanlanganda natija:

*max=55*

**Nazorat savollari**

1. Dinamik ma’lumotlar tuzilmasi nima va uning statik tuzilmalardan afzalligini tushuntiring?
2. Ro’yhat tuzilmasi nima va ro’yhatning qanday turlarini bilasiz?
3. Ro’hat tuzilmasini dasturda ifodalash qanday amalga oshiriladi?
4. Ro’hat tuzilmasi ustida amal bajarish algoritmlarini tushuntiring
5. Ikki bog’lamli ro’yhat nima va uni bir bog’lamli ro’hatdan afzalligi va kamchiligini tushuntiring.

## Topshiriq

Variantlar:

1. Elementni n pozitsiyaga siljitish dasturini tuzing.
2. Ro’yhat nusxasini yarating.
3. Ro’yhat boshiga element qo’yish.
4. Ikkita ro’yhat birlashtirilsin.
5. Ro’yhatning n-inchi elementi o’chirilsin.
6. Ro’yhat n-inchi elementidan keyin yangi element qo’yilsin.
7. Ikkita ro’yhat umumiy elementlaridan tashkil topgan ro’yhat yaratilsin.
8. Ro’yhat elementlari o’sish tartibida joylashtirilsin.
9. Ro’yhat har ikkinchi elementi o’chirilsin.
10. Ro’yhat har uchinchi elementi o’chirilsin.
11. Ro’yhat elementlari kamayish tartibida joylashtirilsin.
12. Ro’yhat tozalansin.
13. Futbol jamosining 20 ta o’yinchilari familiyalaridan tashkil topgan halqasimon ro’yhat berilgan. O’yinchilar 2 ta guruhga 10 tadan ajratilsin. Ikkinchi guruhga umumiy o’yinchilarni har 12-inchisi kirsin.
14. Sportchi familiyalaridan tashkil topgan ikkita halqasimon ro’yhat berilgan. Qura tashlash amalga oshirilsin. Birinchi guruhdagi har n-inchi sportchi, ikkinchi guruhdagi har m-inchi sportchi bilan raqib bo’lsin.
15. Lotoreya ishtirokchilari familiyalari va mukofotlar nomlaridan tashkil topgan 2 ta halqasimon ro’yhat berilgan. N ta ishtirokchi g’olib bo’lsin (har K-inchi). Mukofotlarni qayta hisoblash soni - t.
16. O’quvchilar familiyalari va imtihon biletlari raqamlaridan tashkil topgan 2 ta halqasimon ro’yhat berilgan. O’quvchilar tomonidan olingan bilet raqamlari aniqlansin. Imtihon biletlari uchun qayta hisoblash soni - E, o’quvchilar uchun esa - K.
17. Mahsulot nomlaridan tashkil topgan ro’yhat berilgan. Ro’yhat elementlaridagi SONY firmasida ishlab chiqilgan mahsulotlardan tashkil topgan yangi ro’yhat yarating.
18. 2 ta guruh talabalari familiyalaridan tashkil topgan 2 ta ro’yhat berilgan. Birinchi guruhdan L ta talaba ikkinchi guruhga o’tkazilsin. Qayta hisoblashlar soni - K.
19. BOSCH va PHILIPS konsernlari tomonidan ishlab chiqilgan mahsulot nomlaridan tashkil topgan ikkita ro’yhat berilgan. Har ikkala firma tomonidan ishlab chiqilgan bir xil mahsulotlar ro’yhati tuzilsin.
20. Futbol jamoasining asosiy va zahira tarkibi o’yinchilari familiyalaridan tashkil topgan ikkita ro’yhat berilgan. K ta o’yinchi almashtirilsin.
21. 1- va 2-vzvod askarlari familiyalaridan tashkil topgan ikkita ro’yhat berilgan. Hujum natijasida 1-chi vzvoddan M ta askar halok bo’ldi. Ikkinchi vzvod askarlaridan birinchi vzvod to’ldirilsin.
22. Mahsulot nomlari va xaridorlar familiyalaridan tashkil topgan ikkita ro’yhat berilgan. Har bir N-chi xaridor M-chi mahsulotni sotib oladi. Xarid

qilingan mahsulotlar ro’yhatini chiqaring.

1. SONY va SHARP firmalari tomonidan ishlab chiqilgan mahsulot nomlaridan tashkil topgan ikkita ro’yhat berilgan. O’zaro raqobat qiluvchi mahsulotlar ro’yhatini tuzing.
2. Talabalar ismlaridan iborat ro’yhat berilgan. Ismining uzunligi eng katta bo’lgan talabani ro’yhat boshiga joylang.
3. Talabalar familiyalaridan iborat halqasimon ro’yhat berilgan. Har k-inchi talabadan 3 tasi ro’yhatdan ajratib olinsin.
4. Talabalar ismlaridan iborat massiv elementlarini berilgan halqasimon ro’yhatning har k-elementidan keyin joylashtiring.
5. 2 ta halqasimon ro’yhatni galma-galdan har 3-elementidan umumiy bitta yangi ro’yhat hosil qiling.
6. 2 ta ro’yhatning bir xil qiymatli elementlaridan yangi halqasimon ro’yhat yarating.
7. 2 ta ro’yhatning bir xil qiymatli elementlarini ro’yhat boshiga o’tkazing.

30.2 ta ro’yhatning bir xil qiymatli elementlarini ro’yhat oxiriga joylashtiring.