**O'ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI**

**MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI**

**FAKULTET:KAMPYUTER INJNERING (KIF)**



**MUSTAQIL ISH**

**FAN: MALUMOT TUZILMALARI VA ALGARITIMLAR**

**VARIANT № 21**

**MAZVU:** Bog'langan ro'yxatlar klassifikatsiyasi, Chiziqli bog‘langan ro‘yhatlarni mantiqiy tasvirlash

Bajardi:GIYOSDDINOV JAVOXIR

Tekshirdi:RAXMANOV ASQAR TAJIBAYEVICH

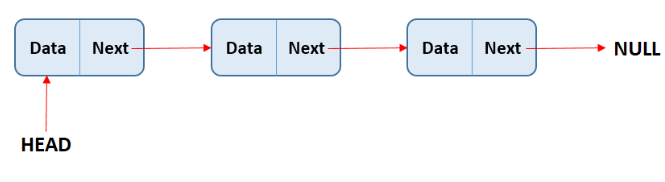
# Mustaqil ish mavzusi: Bog'langan ro'yxatlar klassifikatsiyasi, Chiziqli bog‘langan ro‘yhatlarni mantiqiy tasvirlash

Bog'langan ro'yxatlar (Linked Lists) — bu ma'lumotlarni saqlashda ishlatiladigan ma'lumotlar tuzilmasi bo'lib, ular ma'lumotlarni ketma-ket ravishda joylashtirish imkonini beradi. Ular ro'yxatdagi har bir elementni (tugunni) o'zining keyingi elementiga bog'lash orqali tashkil etiladi. Chiziqli bog'langan ro'yxatlar, odatda, ma'lumotlarni ketma-ket tartibda saqlashda qo'llaniladi, bu esa ularni o'qish va o'zgartirishda samarali qiladi.  
  
Chiziqli bog'langan ro'yxatlar oddiy va keng tarqalgan ma'lumotlar tuzilmasi bo'lib, ular ko'plab dasturlar va algoritmlarda asosiys o'rin tutadi

## **Bog'langan Ro'yxatlar Turlari**

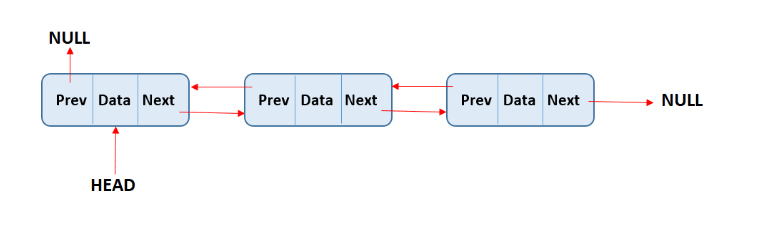
### 1.1 Yagona Bog'langan Ro'yxat (Singly Linked List)

Yagona bog'langan ro'yxat — bu ro'yxatning har bir tuguni faqat bitta ko'rsatkichi (`next`)ga ega bo'lib, bu ko'rsatkich keyingi tugunga bog'lanadi. Ro'yxatning oxirgi tugunining `next` ko'rsatkichi `nullptr` ga teng bo'ladi, bu esa ro'yxatning oxirini bildiradi.



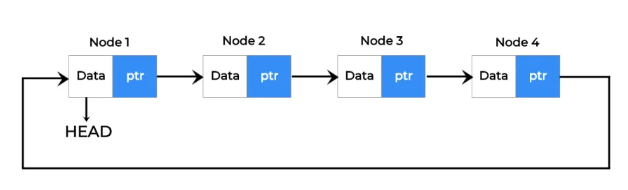
### 1.2 Ikki Tomonlama Bog'langan Ro'yxat (Doubly Linked List)

Ikki tomonlama bog'langan ro'yxatda har bir tugun ikkita ko'rsatkichi (`next` va `prev`)ga ega. `next` ko'rsatkichi keyingi tugunga, `prev` ko'rsatkichi esa oldingi tugunga bog'lanadi. Bu turdagi ro'yxatlar ikki yo'nalishda harakat qilish imkonini beradi.



### 1.3 Doimiy Bog'langan Ro'yxat (Circular Linked List)

Doimiy bog'langan ro'yxatda oxirgi tugun bosh tugunga bog'lanadi. Bu ro'yxatda tugunlar o'zaro doimiy ravishda bog'lanib, aylana hosil qiladi

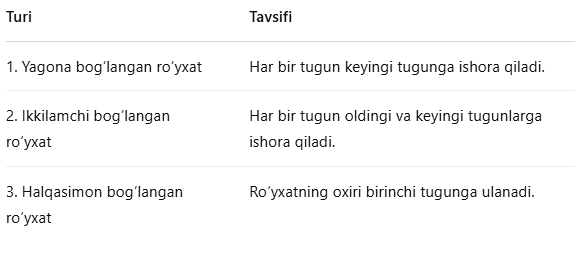


Chiziqli bog‘langan ro‘yxatlarda Stack va Queue tuzilmalarining o‘rni va ahamiyati

### **Klassifikatsiya nima?**

**Klassifikatsiya** — bu narsalarni **ma’lum belgilariga yoki xususiyatlariga qarab guruhlarga ajratish** jarayonidir. Kompyuter fanida, ayniqsa **ma’lumotlar tuzilmalari**da klassifikatsiya deganda **turli xil tuzilmalarni turlariga ajratish** tushuniladi.

**Bog‘langan ro‘yxatlar klassifikatsiyasi** – bu bog‘langan ro‘yxatlarni **ularning tuzilishiga va ishlash tamoyiliga qarab** quyidagi turlarga ajratish demakdir:



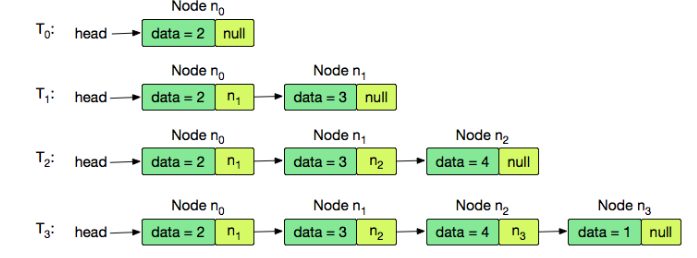
### **Mantiqiy tasvirlash nima?**

**Mantiqiy tasvirlash** — bu ma’lumotlar tuzilmasining **ichki tuzilishini so‘z bilan yoki chizma orqali tushuntirish**. Ya’ni, qanday elementlar bor, ular bir-biri bilan qanday bog‘langan — shularni ko‘rsatish

### Chiziqli bog‘langan ro‘yxatning mantiqiy tuzilishi

Chiziqli bog‘langan ro‘yxatda har bir **tugun (node)** ikki qismdan iborat bo‘ladi:

1. **Ma’lumot maydoni (data)** – bu foydalanuvchidan olingan qiymat, masalan: 5, 10, 3...
2. **Ko‘rsatkich (pointer / next)** – bu keyingi tugunning manzilini ko‘rsatadi.



**mantiqiy tasvirlash** bu — ro‘yxat tuzilishini oddiy so‘zlar va chizmalar yordamida ko‘rsatishdir.

# Xulosa:

Bog'langan ro'yxatlar – ma'lumotlarni tizimli ravishda saqlashda ishlatiladigan muhim ma'lumotlar tuzilmasidir. Ular nafaqat oddiy, balki murakkab tuzilmalarda ham ma'lumotlarni boshqarishda qo'llaniladi. Yagona bog'langan ro'yxatlar oddiy va samarali bo'lsa, ikki tomonlama bog'langan ro'yxatlar ko'proq moslashuvchanlikni ta'minlaydi. Doimiy bog'langan ro'yxatlar esa doimiy aylanishni ta'minlaydi.

**MASALA:**

Talabalar ro‘yxatini bog‘langan ro‘yhat (linked list) yordamida yaratish. Har bir talabada quyidagi ma'lumotlar bo‘lsin

DASTUR\_KODI:

#include <iostream>

#include <string>

using namespace std;

struct talaba{

string fi;

int id;

talaba\* next;

};

talaba\* bosh = NULL;

void talaba\_qosh(string fi,int id){

talaba\* yengi = new talaba;

yengi->fi=fi;

yengi->id=id;

yengi->next = NULL;

talaba\* a = bosh;

if(bosh == NULL){

bosh = yengi;

return;

}

while(a->next!=NULL){

a = a->next;

}

a->next=yengi;

}

void print(){

talaba\* a = bosh;

while(a!=NULL){

cout<<"F.I.SH: "<<a->fi<<", ID: "<<a->id<<endl;

a = a->next;

}

}

int main(){

talaba\_qosh("javoxir",12);

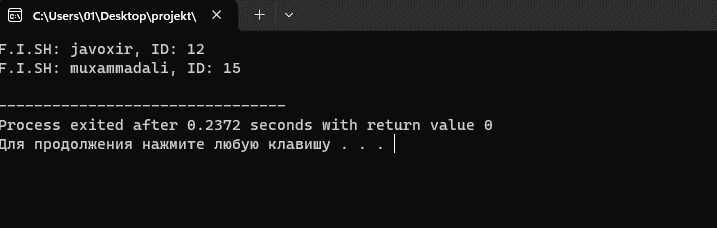
talaba\_qosh("muxammadali",15);

print();

return 0;

}

NATIJA:



#include <iostream>

using namespace std;

struct Node {

int data;

Node\* prev;

Node\* next;

};

Node\* head = nullptr;

Node\* tail = nullptr;

void addToEnd(int value) {

Node\* newNode = new Node();

newNode->data = value;

newNode->prev = nullptr;

newNode->next = nullptr;

if (head == nullptr) {

head = tail = newNode;

} else {

tail->next = newNode;

newNode->prev = tail;

tail = newNode;

}

}

void printForward() {

cout << "Oldindan orqaga: ";

Node\* temp = head;

while (temp != nullptr) {

cout << temp->data << " ";

temp = temp->next;

}

cout << endl;

}

void printBackward() {

cout << "Orqadan oldinga: ";

Node\* temp = tail;

while (temp != nullptr) {

cout << temp->data << " ";

temp = temp->prev;

}

cout << endl;

}

int main() {

int n, value;

cout << "Nechta element kiritmoqchisiz? ";

cin >> n;

for (int i = 0; i < n; i++) {

cout << i+1 << "-sonni kiriting: ";

cin >> value;

addToEnd(value);

}

printForward();

printBackward();

return 0;

}

