

### 13 - AMALIY MASHG'ULOT. XESH JADVALLARI.

**Ishdan maqsad:** Talabalarga Hash jadvallar bo'yicha ma'lumotlar berish va bilimni oshirish. C++ da hash funksiyalarga oid masalalarning dasturini tuzish.

**Nazariy qism:** Xesh jadvallar (hash tables) ma'lumotlarni (qatorda) o'rganilgan tartibda emas, balki biron bir tartibda saqlaydigan, o'ziga xos struktura. Bu struktura ma'lumotlar katta bo'lsa ham, ularga tezkor qidirish imkoniyatini ta'minlaydi. Xesh jadvallar ma'lumotlar keshlashda va ko'p ustunlar, tarjimalarda, tartiblashda va boshqalarda keng qo'llaniladi.

Xesh jadvallarining ba'zi asosiy xususiyatlari quyidagilardir:

1. **Tezkor qidirish:** Xesh jadvallar tezkor qidirishni ta'minlaydi. Agar katta jadvallar yoki katta ma'lumotlar keshlari (hash maps) mavjud bo'lsa, ma'lumotlar qidirish operatsiyalari o(1) vaqt kompleksligida ishlaydi.
2. **Ko'p ma'lumotlar keshi:** Xesh jadvallar ma'lumotlar keshlash uchun yaxshi platformadir. Ular, ma'lumotlarni yadda tutish uchun keng qo'llaniladi, ammo kerak emas bo'lgan ma'lumotlarni ko'p joy egallashadi.
3. **Ko'p xil turdagi ma'lumotlarni o'z ichiga oladi:** Xesh jadvallar ko'p turdagi ma'lumotlarni o'z ichiga oladi, masalan, har qanday ma'lumotlar (matnlar, sonlar, obyektlar) uchun foydalanilishi mumkin.
4. **Xotira ishlatish:** Xesh jadvallar ma'lumotlar yozish va o'qish operatsiyalarida keng qo'llaniladi, ammo ular xotira qulayroq o'zgarishlarni ta'minlaydi.
5. **To'g'ri tartib:** Xesh jadvallar xatolikni oldini olishda yordam beradi va ma'lumotlarni to'g'ri tartibda saqlashda yordam beradi.

Xesh jadvallar (hash tables) ma'lumotlar strukturalarining bir turidir va ularni yaratish uchun quyidagi bosqichlarni amalga oshirish kerak:

**Jadvallar (arrays) yaratish:** Birinchi navbatda, ma'lumotlarni saqlash uchun o'lchamli jadvallar (arrays) yaratish kerak. Bu jadvallar, xotiraning bir qismiga bo'lgan erishish uchun mo'ljallangan sahifalar sifatida xizmat qiladi.

**Hash funksiyasini tanlash:** Hash funksiyasi ma'lumotlarni jadvallarning mos kelishi uchun biron bir ma'lumotni tartiblash uchun yordam beradi. Hash funksiyasi odatda ma'lumotni katta raqamlarga aylantiradi va uchraydigan qiymatni jadvallarning indeksi sifatida ishlatadi.

**Hash funksiyasini ishtirok etuvchi jadvallar tuzish:** Hash funksiyasi yordamida ma'lumotni saqlash uchun mos keluvchi jadvallar tuziladi. Har bir ma'lumotning hash qiymati indeks sifatida ishlatiladi va ma'lumotlar shu indeksga joylashadi.

**Koliziyalarni bartaraf qilish:** Koliziya yuz berishi mumkin, ya'ni ikkita yoki undan ko'p ma'lumotning bir xil hash qiymatiga ega bo'lishi. Ushbu holatda, moslikni saqlash uchun

qo'llaniladigan qo'llab-quvvatlovchi algoritmlar yoki jadvallar ichida xil ma'lumotlar ro'yhati yaratilishi kerak.

**Ma'lumotni Jadvallarga Joylashgan Indeksni Topish va Ma'lumotni Saqlash:** Hash funksiyasi yordamida ma'lumotning jadvallarga joylashgan indeksini topish va shu indeksda jadvallarga ma'lumotni saqlash.

### **Amaliy qism:**

Quyidagi C++ kodida oddiy bir xesh jadvalli yaratilishi ko'rsatilgan:

```
#include <iostream>
#include <vector>
using namespace std;
// Hash funksiyasi
int hashFunction(int key, int size) {
    return key % size; // Odatiy hash funksiyasi: keyingi bo'limni indeks
sifatida qo'llash
}
// Xesh jadvalli strukturasini tuzish
class HashTable {
private:
    int size;
    vector<int> *table;
public:
    // Konstruktor
    HashTable(int size) {
        this->size = size;
        table = new vector<int>[size];
    }
    // Ma'lumotni qo'shish
    void insert(int key) {
        int index = hashFunction(key, size);
        table[index].push_back(key);
    }
    // Ma'lumotni izlash
    bool search(int key) {
        int index = hashFunction(key, size);
        for (int i = 0; i < table[index].size(); ++i) {
            if (table[index][i] == key) {
                return true;
            }
        }
    }
}
```

```

        return false;
    }
};

int main() {
    // Xesh jadvalli yaratish
    HashTable hashTable(10);
    // Ma'lumotlarni qo'shish
    hashTable.insert(5);
    hashTable.insert(15);
    hashTable.insert(25);
    // Ma'lumotlarni izlash
    cout << "5 ma'lumoti topildimi? " << (hashTable.search(5) ? "Ha" :
    "Yo'q") << endl;
    cout << "10 ma'lumoti topildimi? " << (hashTable.search(10) ? "Ha" :
    "Yo'q") << endl;
    return 0;
}

```

### 1-misol: Ma'lumotlarni jadvallardan olish.

```

// Ma'lumotni olish uchun funksiya
int retrieve(int key) {
    int index = hashFunction(key);
    if (!table[index].empty()) {
        for (int i = 0; i < table[index].size(); ++i) {
            if (table[index][i] == key) {
                return table[index][i];
            }
        }
    }
    return -1; // Ma'lumot topilmadi
}

```

### 2-misol: Ma'lumotni o'chirish.

```

// Ma'lumotni o'chirish uchun funksiya
void remove(int key) {
    int index = hashFunction(key);
    if (!table[index].empty()) {
        for (int i = 0; i < table[index].size(); ++i) {
            if (table[index][i] == key) {
                table[index].erase(table[index].begin() + i);
                cout << "Ma'lumot muvaffaqiyatli o'chirildi!" << endl;
                return;
            }
        }
    }
}

```

```

    }
  }
}
cout << "Ma'lumot topilmadi, o'chirish amalga oshmadi!" << endl;
}

```

### **Mustaqil bajarish uchun topshiriqlar**

1. Xesh jadvallarini (hash tables) yaratib, ma'lumotlarni qo'shish va izlash funksiyalarini qo'llash.
2. Koliziyalarni boshqarish usullarini qo'llab-quvvatlash va ularni test qilish.
3. Xesh jadvallarining kattaligini o'lchab chiqish uchun kattalikni qo'shish va o'chirish jarayonlarini amalga oshirish.
4. Xesh jadvallaridan ma'lumotlarni izlash va ularga kirish.
5. Xesh jadvallariga ma'lumotlarni qo'shish va ularni yangilash.
6. Kriptografik xesh jadvallarini ishlatib, ma'lumotlarni qo'shish va izlash.
7. Xesh jadvallarining xavfsizlik darajasini tekshirish uchun bir nechta ma'lumotlar kiritish.
8. Xesh jadvallarining ma'lumotlar qo'shish va izlash jarayonlarining tezligini o'lchab chiqish.
9. Matnlarni xesh jadvallariga saqlash va ularni izlash.
10. Sonlarni xesh jadvallariga ma'lumotlarni qo'shish va ularni test qilish.
11. Obyektlarni xesh jadvallariga qo'shish va ularni test qilish.
12. Xesh jadvallarining kattaligini o'lchab chiqish uchun kattalikni qo'shish va o'chirish jarayonlarini test qilish.
13. Xesh jadvallarining to'g'ri ishlashini tekshirish uchun xodimlar qo'shish va ularni test qilish.
14. Xesh jadvallarida kichik matnlar uchun saqlash va ularni test qilish.
15. Oddiy xesh jadvallarining ishtirokida ma'lumotlarni qo'shish va izlash jarayonlarini test qilish.