14 - AMALIY MASHG‘ULOT. *HASH FUNCTION.*

Ishdan maqsad: Talabalarga Hash funksiyalar bo‘yicha ma‘lumotlar berish va bilimini oshirish. C++ da hash funksiyalarga oid masalalarning dasturini tuzish.

Nazariy qism: Hash funksiyalar - malumotlar dasturdagi ma’lumotlar to’plamini o’ziga xos kichik ma’lumotlar ketma-ketligiga (hash value yoki hash code deb ataladi) o’zgartiruvchi funksiyalardir. Bu funksiyalar ma’lum bir uzunlikdagi ma’lumotlar ketma-ketligini qabul qilib, unga mos bo’lgan kichik ma’lumotlar ketma-ketligini hosil qiladi.

Hash funksiyalari ko’pincha to’liq yoritilgan katta ma’lumotlarni kichik korsatkichga aylantirish uchun ishlatiladi. Misol uchun, haqiqiy o’zbekcha so’zni ko’plab to’plamlar yoki kitoblar ichida qidirib topishda qulaylik bilan ishlatiladi.

Bu funksiyalar quyidagi xususiyatlarga ega bo’lishi kerak:

1. ***Unikallik:*** Har bir kiritilgan ma’lumot uchun unikal (faqatgina bir biriga xos) hash qiymat hosil qilishi kerak.

2. ***Tezlik***: Hash funksiyasi tezkor ishlashi kerak. Bu, katta ma’lumotlarga ishlov berishda muhimdir.

3. ***Keraklilik:*** Ma’lumotlarni hosil qilish vaqti keyingi ma’lumotlarni hash qilish vaqtidan qisqa bo’lishi kerak.

4. ***Qaror qilinuvchi (Determinastik):*** Bir xil ma’lumotlarni hash qilish uchun har doim bir xil hash qiymat hosil bo’lishi kerak.

Hash funksiyalari kriptografiyada, ma’lumotlar bazalarida, ma’lumotlar matrislarida, kalit so’zlarda, tarjimalarda va boshqalar kabi ko’plab sohalarda qo’llaniladi.

Ba’zi mashhur hash funksiyalari:

1. ***MD5:*** 128-bit (16 byte) hash qiymat hosil qiladi.
2. ***SHA-1:*** 160-bit (20 byte) hash qiymat hosil qiladi.
3. ***SHA-256:*** 256-bit (32 byte) hash qiymat hosil qiladi.
4. ***SHA-512:*** 512-bit (64 byte) hash qiymat hosil qiladi.

Bu funksiyalar quyidagi muammolar bilan ajratiladi:

1. *Kichiklik:* Ma’lumotlarni kichik korsatkichga aylantirishda hashning kichikliki muhimdir. Ba’zi funksiyalar kichik ma’lumotlar uchun yaxshi emas.

2. *Tezlik:* Hash funksiyasi ishlovchi vaqtini ko’paytirish haqida yondashuvlar mavjud.

3. *Xavfsizlik:* Kriptografiyada ishlatilgan hash funksiyalari xavfsizligi muhimdir. Ba’zi eski funksiyalar, masalan, MD5, xavfsizlik muammolariga duch kelishi mumkin.

Yangi SHA funksiyalari keng tarqalgan ma'lumotlar tahlili (cryptographic hash function) sifatida yaxshi tanilgan.

Hash funksiyalari juda keng qo'llaniladigan algoritmlardir va dastlabki maqsad ma'lumotlarni boshqa ma'lumotlarga taalluqli korsatkichlarga aylantirishdir.

Amaliy qism:

1-masala: Berilgan matnlarning MD5 hash qiymatlarini solishtirish.

```
#include <iostream>
#include <openssl/md5.h>
#include <string>
#include <unordered_map>
std::string md5_hash(const std::string& input) {
    unsigned char result[MD5_DIGEST_LENGTH];
    MD5((const unsigned char*)input.c_str(), input.length(), result);
    std::string hash;
    char buffer[3];
    for (int i = 0; i < MD5_DIGEST_LENGTH; ++i) {
        sprintf(buffer, "%02x", result[i]);
        hash += buffer;
    }
    return hash;
}
int main() {
    std::unordered_map<std::string, std::string> texts_to_hashes;
    texts_to_hashes["Hello"] = md5_hash("Hello");
    texts_to_hashes["World"] = md5_hash("World");
    texts_to_hashes["Hello, World!"] = md5_hash("Hello, World!");
    // Solishtirish
    for (const auto& pair : texts_to_hashes) {
        std::cout << "Matn: " << pair.first << "\tHash: " << pair.second <<
std::endl;
    }
    return 0;
}
```

Bu dastur berilgan matnlarning MD5 hash qiymatlarini hisoblaydi va ularni solishtiradi. Natijada, matn va uning MD5 hash qiymati chiqadi.

2-masala: Matnlar to'plamining SHA-1 hash qiymatini hisoblash

```
#include <iostream>
#include <openssl/sha.h>
```

```
#include <string>
std::string sha1_hash(const std::string& input) {
    unsigned char result[SHA_DIGEST_LENGTH];
    SHA1((const unsigned char*)input.c_str(), input.length(), result);
    std::string hash;
    char buffer[3];
    for (int i = 0; i < SHA_DIGEST_LENGTH; ++i) {
        sprintf(buffer, "%02x", result[i]);
        hash += buffer;
    }
    return hash;
}
int main() {
    std::string input1 = "Hello";
    std::string input2 = "World";
    std::string concatenated = input1 + input2;
    std::string hash = sha1_hash(concatenated);
    std::cout << "SHA-1 Hash qiymati: " << hash << std::endl;
    return 0;
}
```

Bu kodlar foydalanuvchidan matn kiritishni so'raydi, keyin kiritilgan matning SHA-1 hash qiymatini hisoblaydi va uni ekranga chiqaradi. Yuqoridagi kodlar OpenSSL kutubxonasidan foydalanadi. Natijada, foydalanuvchidan kiritilgan matning yoki matnlar to'plamining SHA-1 hash qiymati chiqadi.

3-masala: Matnlar to'plamining SHA-256 hash qiymatini hisoblash.

```
#include <iostream>
#include <openssl/sha.h>
#include <string>
std::string sha256_hash(const std::string& input) {
    unsigned char result[SHA256_DIGEST_LENGTH];
    SHA256_CTX sha256;
    SHA256_Init(&sha256);
    SHA256_Update(&sha256, input.c_str(), input.length());
    SHA256_Final(result, &sha256);
    std::string hash;
    char buffer[3];
    for (int i = 0; i < SHA256_DIGEST_LENGTH; ++i) {
        sprintf(buffer, "%02x", result[i]);
        hash += buffer;
    }
```

```
    return hash;
}
int main() {
    std::string input;
    std::cout << "Matn kiriting: ";
    std::getline(std::cin, input);
    std::string hash = sha256_hash(input);
    std::cout << "SHA-256 Hash qiymati: " << hash << std::endl;
    return 0;
}
```

Mustaqil bajarish uchun topshiriqlar

1. Kiritilgan matnning MD5, SHA-1 yoki SHA-256 hash qiymatlarini hisoblang va ekranga chiqaring.
2. Kiritilgan matnning hash qiymatini hisoblang va ekranga chiqaring.
3. Kiritilgan matnni boshqarish uchun to'g'ri kichik xarflarga o'tkazing va keyin hash qiling va natijani ekranga chiqaring.
4. Bir nechta matnlarni kiriting, keyin har bir matnning hash qiymatini hisoblang va ro'yxatni ekranga chiqaring.
5. Bir nechta matnlar kiriting, keyin ularning hash qiymatlarini hisoblang va ularni solishtiring. Agar ular bir xil bo'lsa, "Barcha hash qiymatlari bir xil" deb ekranga chiqaring.
6. Kiritilgan matning SHA-256 yoki SHA-512 hash qiymatlarini hisoblang va ekranga chiqaring.
7. Fayl nomini kiriting, keyin faylning o'lchami bo'lgan ma'lumotlarni hash qiling va natijani ekranga chiqaring.
8. Bir nechta katta matnlarni kiritib, ularni hash qilish va ularning ishlovchi vaqtini o'lchab chiqaring.
9. Bir nechta foydalanuvchiga o'zgaruvchilarni kiritib, ularning hash qiymatlarini hisoblang va eng yuqori hash qiymatiga ega bo'lgan foydalanuvchi nomini ekranga chiqaring.
10. Bir nechta matnlar ro'yxatini kiritib, ularning o'zaro mosligini tekshirish uchun hash funksiyalaridan foydalanuvchi dasturini tuzing.