**1fd75c3872a94e26ad68c7fa7667bdc82c07dd4ba85f4a6793f7a2b4e943b8fd**

**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU

**ÖDEV BAŞLIĞI**

**G181210601 – Muhammed BÜYÜKKESKİN**

**G191210100 – Utku BİLGİN**

**B191210556 – Hana KAJAN**

**SAKARYA**

**Ağustos, 2021**

Programlama Dillerinin Prensipleri Dersi

Luhn algoritması

1. Projenin konusu

Luhn, bir kontrol anahtarı aracılığıyla numaraları (kredi kartı, SIRET, vb.) kontrol etmeyi mümkün kılar. Tek karakterde bulunan hatalar bile algoritma tarafından tespit edilebilir.

Luhn, MasterCard, American Express (AMEX), Visa ve tüm kredi kartları tarafından kullanıldığı için bilinir.

LUHN formül algoritması, 1954 yılında IBM’de araştırmacı olarak çalışırken Hans Peter Luhn adlı bir Alman Bilgisayar Bilimcisi tarafından geliştirilmiştir.  Algoritmanın kesin işleyişi, 19. yüzyılın başlarında Carl Friedrich Gauss tarafından geliştirilen matematiksel bir teknik olan modüler aritmetiğe dayanmaktadır. Ayrıntılı çalışmaları oldukça karmaşık olsa da, bilgisayarların müşteriler tarafından sağlanan kredi kartı numaralarının doğru olup olmadığını hızlı bir şekilde değerlendirmesine izin vermesiyle bilinir.

Bunu yapmanın yolu, verilen kredi kartı numarasına bir dizi hesaplama uygulamak, bu hesaplamaların sonuçlarını toplamak ve elde edilen sayının beklenen sonuçla eşleşip eşleşmediğini kontrol etmektir. Varsa, kredi numarası geçerli sayılır. Aksi takdirde, algoritma kredi kartı numarasını reddederek kullanıcının numarayı girerken bir hata yaptığını gösterir.

Luhn Algoritması içindeki temel kavramlardan biri, sözde “kontrol basamakları” nın kullanılmasıdır. Bu basamaklar, tüm sayının gerçek olup olmadığını doğrulamaya veya “kontrol etmeye” yardımcı olmak için daha geniş sayı dizisine eklenen sayılardan oluşur.

Günümüzde Luhn Algoritması, popüler programlama dillerine ve kod kitaplıklarına entegre edilmiştir ve bu, Luhn tabanlı kimlik numarası doğrulamasını yeni yazılım uygulamalarına dahil etmeyi nispeten kolaylaştırmaktadır.

1. **Kullanım yerleri**

Özel rakam dizilimlerinden meydana gelen kredi kartları kullanım alanlarının artması ve çok fazla ilgi görmesi nedeniyle dolandırıcıların ilgi odağı haline gelmeye başlamıştır. Her kredi kartına kendisine ait özel numaralar oluşturularak dolandırıcılıkların önüne geçmek ve kişisellik yaratılmak istenilmiştir. Bu noktada Hans Peter LUHN tarafından ortaya çıkarılan LUHN algoritması kredi kartlarının kendilerine özel numaralarını kullanarak doğruluğunu sorgulama konusunda kullanılmaya başlamıştır. Bu algoritma günümüzde bankalar, online alışveriş siteleri vb. gibi birçok alanda kullanılmaktadır. Bu algoritmanın avantajları olduğu gibi dezavantajları da mevcuttur. Başlıca avantaj ve dezavantajları aşağıda olduğu gibidir;

1. **Luhn algoritmasının avantajları ve dezavantajları**

**Avantajları:**  
- Tek hanede yapılan hataların bile tespiti mümkündür.

* Kredi kartı ile ödemede güvenlik yardımcısıdır. Kart numarasının doğruluğunu tespit eder.
* Hızlı çalışır.
* Basit çalışma mantığına sahiptir.

**Dezavantajları :**

* Herhangi bir kriptolamaya sahip değildir.
* Sadece kart numarasının doğruluğunu kontrol eder. Kart iptal edilmiş veya limit yetersizliği durumlarını kontrol edemez.

Kredi kartları için tek başına onay elemanı olarak kullanılamaz.

1. **Proje grubunun görev paylaşımı ve projenin nasıl yapıldığı**
2. Hana'nın görevi : rapor oluşturma hazırlama işleri, Program.c kodları, Proje geliştirilebilirlik araştırmaları, Genel kontrol
3. Utku'nun görevi : github yönetimi, program.h ,makefile yapısı, yönetim, görüşmelerin ayarlanması (google meet), deneme dosyalarının hazırlanması ve random sayı üretme fonksiyonları, Proje geliştirilebilirlik araştırmaları, Genel kontrol
4. Muhammed'in görevi : Test.c programını, luhn algoritmasının avantaj dezavantajlarını bulamak ve luhn algoritmasının kullanım alanları, Proje geliştirilebilirlik araştırmaları, Genel kontrol

**Proje yapımında izlenen yol haritası aşağıda belirtildiği gibidir.**

1) Grup üyeleri tarafından proje ile ilgili araştırmalar yapıldı.

2) Google meet üzerinden grup üyelerinin katılımıyla toplantı gerçekleştirildi.

3) Görev paylaşımları yapıldı.

4) Sadece grup üyelerinin erişimine açık bir Github deposu oluşturuldu.

5) Yapılan toplantılar ve yazışmalar ile kodlar istişare yapılarak yazıldı.

6) Yazılan kodlar program bütünlüğünü sağlamak için birleştirildi.

7) Test dosyası yazıldı.

8) Kodlar için gerekli deneme dokümanları yazıldı.

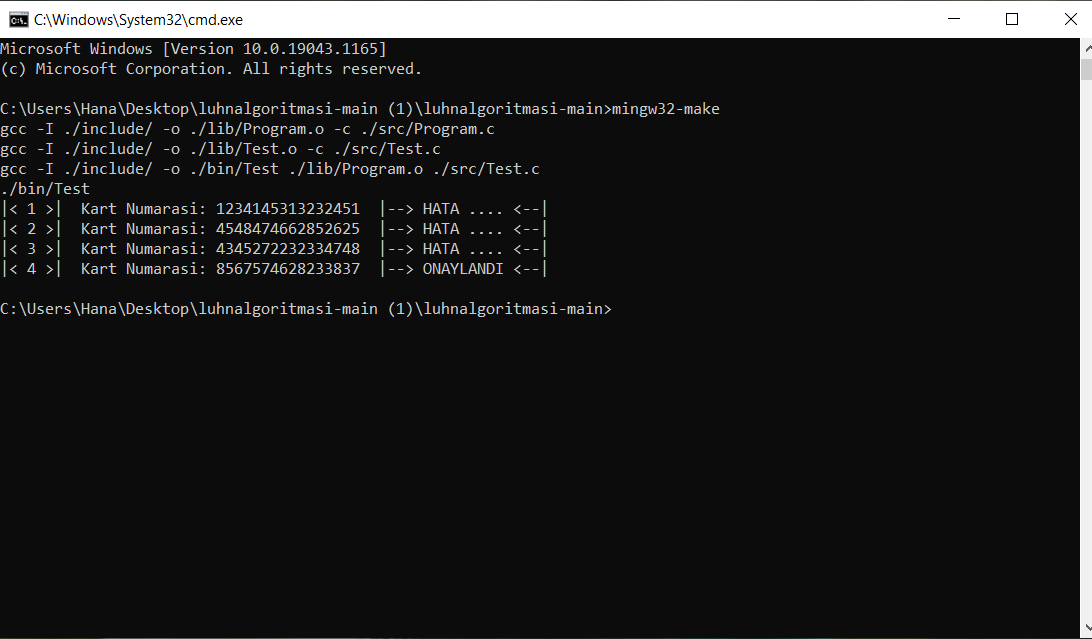
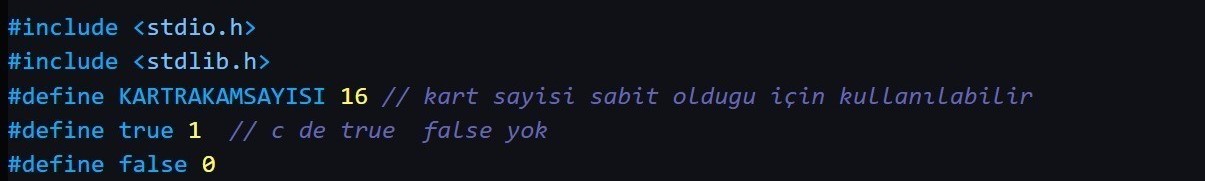
9) Makefile yapısı yazıldı.

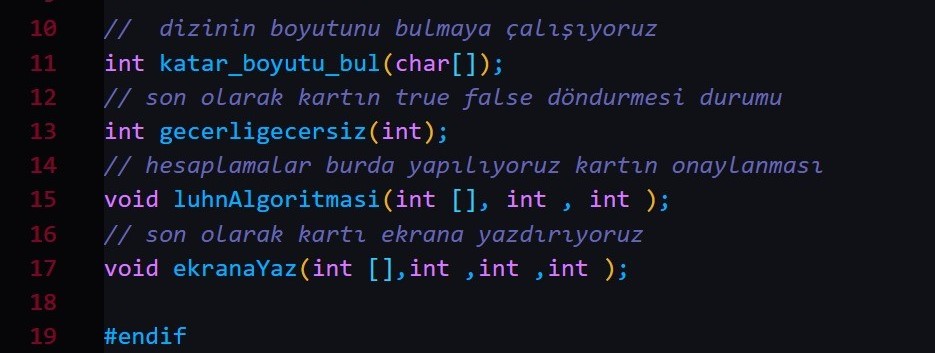
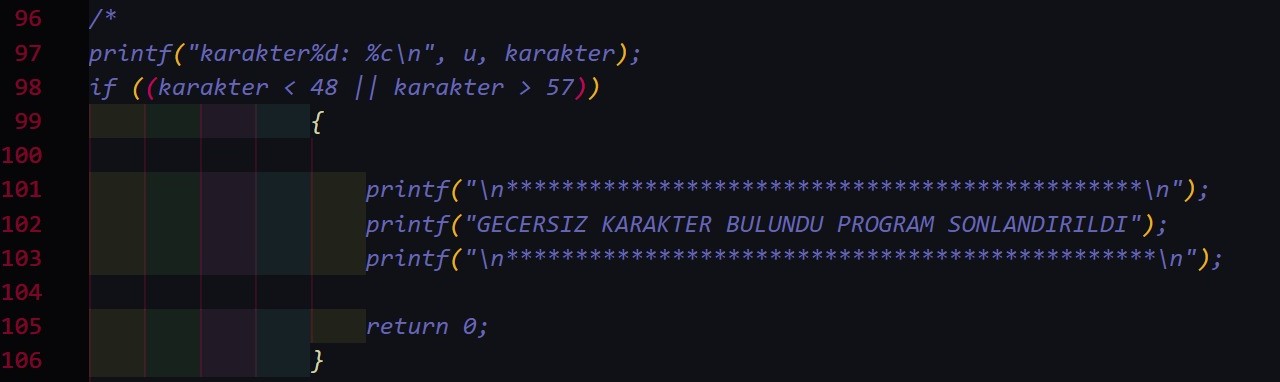
10)Proje derlendi ve test edildi. Faal olduğu görüldü.

1. **Projenın eksik kalan yönleri**

Bu projede eksik olan tek bir nokta vardır. O da dosya okuma sırasında dışarıdan veri aldığımızda gelecek verilerin içinde harflerin ya da int e çevrilmeyen karakterler bulunması kontrolu. Bu konuda gelecek veriler ve geleceği dosya yapısı belli olduğu için ve yapamadığımız için projeye eklenmedi. Ama onunla ilgili çalışmalarımız mevcuttur. Proje dokümanında mevcuttur.

1. Çıktılar



* C dilinde true false bulunmadığı için tanımlamalar yapıldı ve bir sabit tanımlandı
*  h kütüphanesi fonksiyonları tanımlandı
* Burada dışarıdan gelen değerler için control yapılmaya çalışıldı

© 2021 Sakarya Üniversitesi.

Bu rapor bizim özgün çalışmamızdır. Faydalanmış olduğumuz kaynakları içeresinde belirttik. Her hangi bir kopya işleminde sorumluluk bize aittir.