Projenizin gerekliliklerini dikkate alarak, bu final proje için aşağıdaki adımları takip edebilirsiniz:

1. **Kod Yorumlama:** Proje için sağlanan C kodunu inceleyin. Ne yaptığını anladığınızdan emin olun ve gerektiği yerde yorumlar ekleyin. Ancak, Samer Buna'nın "The Mistakes I Made…" adlı eserindeki önerilere dikkat edin, özellikle "Writing Comments About the Obvious Things" bölümüne dikkat edin. Yorumlarınızın anlamlı ve değerli olmasını sağlayın.
2. **Test Durumları Oluşturma:** İkinci aşama, kodun doğru çalıştığını doğrulamak için test durumları oluşturmayı içerir. Her fonksiyon veya işlevsellik için bir dizi test durumu oluşturun ve her birini belgelendirin. Buna, test durumunun ne olduğu, beklendiği çıktının ne olduğu ve gerçek çıktının ne olduğunu içeren bilgiler dahildir.
3. **Testleri Çalıştırma:** Oluşturulan test durumlarını çalıştırın ve sonuçları belgelendirin. Eğer bir test durumu başarısız olursa, hatayı ve olası çözümü belirtin.
4. **MS Teams Kullanımı:** Tüm işbirliği MS Teams üzerinden yapılmalıdır. Bu platform, projenin tüm yönlerini organize etmek için kullanılabilir. İşbirliği süreci, proje notlandırması için önemli bir unsurdur.
5. **Proje Planı:** Tüm takım üyeleri, bir proje planı oluşturmalı ve sürdürmelidir. Bu plan, projenin tüm adımlarını içermeli ve projenin nasıl tamamlanacağını belirtmelidir. Planın oluşturulması ve sürdürülmesi, projenin notlandırılmasının bir parçasıdır.
6. **Son Kullanıcı Deneyimi:** Projenizin bir parçası olarak, son kullanıcı deneyimini düşünün. Kodunuzun kullanımı kolay ve anlaşılır olmalıdır.
7. **Kaynak Kontrolü:** Projenin her aşamasında kaynak kontrolünü kullanın. Bu, kodunuzun her versiyonunu izlemenize ve gerektiğinde önceki versiyonlara geri dönmenize olanak sağlar.

Bu adımları takip etmek, projenizin başarılı olmasına yardımcı olacaktır. Unutmayın, projenin ana hedefi, Yazılım Geliştirme Yaşam Döngüsü'nün (SDLC) geç aşamalarını anlama ve uygulama yeteneğinizi göstermektir.

Bu ödevin, dört farklı C modülü üzerinde çalışmayı içerdiği anlaşılıyor: Fundamentals, Manipulations, Tokenizing, ve Conversions. Her biri belirli bir konuyu ele alıyor ve bu konularla ilgili belirli fonksiyonları içeriyor.

Projenizin genel adımlarını aşağıdaki gibi belirleyebiliriz:

1. **Fonksiyonları Anlama:** Her modüldeki fonksiyonların ne yaptığını anladığınızdan emin olun. Eğer bir fonksiyonun amacını veya kodun ne yaptığını anlamıyorsanız, öncelikle bunu anlamak için araştırma yapın.
2. **Kod Yorumlama:** Kodu yorumlamak, başkalarının kodunuzu okumasını ve anlamasını kolaylaştırır. Ancak, her şeyi yorumlamak yerine, sadece kodun karmaşık veya belirsiz kısımlarını yorumlayın.
3. **Test Durumları Oluşturma:** Her fonksiyon için bir dizi test durumu oluşturun. Test durumları, fonksiyonların doğru şekilde çalıştığını doğrulamanın bir yoludur.
4. **Testleri Çalıştırma ve Sonuçları Belgelendirme:** Her test durumunu çalıştırın ve sonuçları belgelendirin. Başarısız olan testlerin neden başarısız olduğunu ve nasıl düzeltilebileceğini belirtin.
5. **İşbirliği Araçları Kullanma:** Tüm işbirliği MS Teams üzerinde gerçekleştirilmeli ve kodlama sürecinde Git gibi bir kaynak kontrol sistemi kullanılmalıdır.

Her bir modülün ve fonksiyonun gereksinimlerini dikkatlice inceleyin ve belirtilen gerekliliklere uygun olarak işlevsellik sağladığınızdan emin olun. Ayrıca, her modül ve fonksiyonun belgelerini oluşturmayı unutmayın. Bu, kodunuzun ne yaptığını açıklar ve başkalarının kodunuzu anlamasını kolaylaştırır.

Son olarak, her bir modül ve fonksiyonun belirtilen araçlarla (gcc, git) derlenebildiğini ve çalıştırılabildiğini doğrulayın.

Bu proje, dört kişilik bir ekip tarafından yürütülecek. Üç programcı modüller A-C'nin tüm yönlerinden sorumlu olacak ve bir programcı / ekip lideri, modül D'yi geliştirmek ve ana uygulamaya entegre etmek için görevlendirilecek.

Ayrıca, projenin farklı versiyonlarının belirli tarihlerde teslim edilmesi gerekiyor. Her versiyon, bir öncekine eklenen yeni işlevselliği içermeli. Bu, kodunuzun sürekli olarak geliştirilmesi ve geliştirilmesi gerektiği anlamına geliyor.

Ek olarak, ekip üyelerinin MS Teams üzerinden iletişim kurmaları ve işbirliği yapmaları gerekiyor. Bu, projenin tüm yönlerinin organize edilmesine yardımcı olur. Ayrıca, tüm proje dosyalarının MS Teams'deki Files klasörünün kökünde bulunması gerekiyor.

C kodu, .png dosyaları olarak sağlanacak ve öğrencilerin bu kodu girmeleri, yorumlamaları, derlemeleri, birim test durumları oluşturmaları ve test sonuçlarını yakalamaları gerekiyor. Versiyonlar 2 ve 3 için, aynı dosya adlarına kod, yorumlar ve testler eklenir veya entegre edilir. Her bir sonraki versiyon, önceki versiyonları içerir.

Versiyon 1, 2 ve 3 için programlama gerekli değildir, sadece sağlanan kaynak kodun girişi ve derlemesi gerekir. Kod, çok az veya hiç giriş doğrulaması içermez ve bu, test notlarıyla ele alınır.

Ayrıca, proje yönetimi planlama şablonu, yorumlama, test etme, test durumu şablonu, git kullanma ve video eğitimlerinin yanı sıra özellikleri belgelenmiş bir dizi belge sağlanacaktır.

Bu projenin değerlendirilmesi çeşitli faktörlere bağlıdır:

1. **Eşit Katkı:** Her öğrenci ekibe eşit katkı sağlamalı veya projenin tüm modüllerinin tamamlanmış bir versiyonunu tek başına sunmalıdır.
2. **Ekip İçi Çalışma:** Aynı ekipteki üyeler genellikle aynı notları alır. Ancak, katkıların farklı olduğu durumlarda (örneğin, sadece iki üye versiyon 3'ü tamamladıysa), bu farklılıkların proje planında net bir şekilde görülebilir olması ve Blackboard gönderim yorumlarında belgelenmiş olması gerekir.
3. **Süre:** Belirlenen tarihlerde gönderimlerin yapılması gerekmektedir. Geç gönderimler, sadece olağanüstü durumlarda ve önceden anlaşıldığında kabul edilir.
4. **Notlar:** Projenin versiyonlarına göre maksimum puanlar belirlenmiştir. Her gönderim niteliksel olarak değerlendirilir ve versiyonuna göre faktörleştirilir.
5. **Ekip İletişimi:** Ekip iletişiminin verimliliği, proje yönetimi ve planlama, belirlenen tarihlerin tutulması, kaynak dosya yorumlarının kalitesi, kapsamlı test sonuçları ve Versiyon 2+ için git kullanım kanıtları da projenin son notunu etkiler.

Genel olarak, bir ekibin tüm öğrencileri her versiyon için aynı notu alacaktır. Ancak, bu kuralın dışında kalan durumlar varsa, bunlar gönderim yorumlarında ve proje planında belirtilmelidir. Ekip projeleri, herkesin çabalarını koordine etme konusunda ekstra çaba gerektirir. Bu, daha yüksek kaliteli ve daha keyifli bir sonuç elde etmeye de yardımcı olabilir. Ancak, bir ekip ile çalışmanın mümkün olmadığı olağanüstü durumlarda, öğretmenle iletişime geçmek gereklidir.

1. Ek A'da belirtilen C dilinde Standart Kütüphane Fonksiyonlarına bakalım:
   * atoi(): Bir stringi (karakter dizisini) tamsayıya dönüştürür.
   * atof(): Bir stringi çift hassasiyetli ondalık sayıya (double) dönüştürür.
   * atol(): Bir stringi uzun tamsayıya (long) dönüştürür.
   * strlen(): Bir stringin uzunluğunu döndürür.
   * strcpy(): Bir stringi başka bir stringe kopyalar.
   * strcat(): İki stringi birleştirir.
   * strcmp(): İki stringi karşılaştırır.
   * strstr(): Bir string içinde başka bir string arar.
   * strtok(): Bir stringi belirli bir ayırıcıya göre tokenlara (parçalara) ayırır.
2. Ek B'de belirtilen proje teslimleri ve süreleri:
   * Her bir ekip üyesi belirli bir modül üzerinde çalışacak ve aşağıdakileri tamamlamalıdır:
     + Modülün kodunu girmeli ve yorumlamalıdır.
     + Modül için kapsamlı birim testleri yapmalı ve test sonuçlarını kaydetmelidir.
     + Test vakası giriş ve çıktılarını gösteren bir metin yakalamalıdır.
     + Yüklenen tüm dosyaları ve güncellenmiş Proje Planını MS Teams kanalında paylaşmalıdır.
   * Ekip lideri:
     + Tüm birim modül dosyaları ile entegre bir program derlemelidir.
     + Her bir modülün temel işlevini göstermek için basit entegrasyon testleri yapmalı ve test sonuçlarını kaydetmelidir.
     + Proje planını güncellemeli ve tüm dosyaları bir ZIP arşivi içine koymalıdır.
3. "Version 2" için:
   * Ekip üyeleri sürüm 1 modül dosyalarını git versiyon kontrol sistemine taşımalıdır.
   * Sürüm 2 gereksinimlerini Sürüm 1 modül dosyalarına eklemelidirler.
   * Her bir ekip üyesi kendi birim modülünün Sürüm 2 dosyalarını eklemeli ve kontrol etmelidir.
4. "Version 3" için:
   * Sürüm 3, sürüm 2 ile aynı süreci takip eder.
   * Sadece projenin başında bir kere yapılması gereken git versiyon kontrol sistemi kurulumu hariç.

Proje tesliminde, tüm dosyaların bir ZIP arşivinde, klasör yapısı olmadan veya iç içe geçmiş zip dosyaları olmadan arşivlenmesi gerekmektedir. Teslimatlar, dönemin son gününe kadar tamamlanmalıdır. Teslimattan sonra ve derslerin son günü arasında yapılan her şey, ek süre maliyetine tabidir.