Bir operatör yerleştirdiğinizde, hücrelerin daha önce bir başkası tarafından işgal edilip edilmediğini kontrol etmeniz gerekir. yerleştirilmiş operatör veya değil. **En az bir hücre** önceden yerleştirilmiş operatör tarafından işgal edildiğinde, bir var **ÇATIŞMA HATASI**. <u>Şekil 5</u>, uyuşmazlık hatalarını örneklemektedir.

Şekil 5: ÇATIŞMA HATASI Örneği

Bir işleci ızgaraya yerleştirirken bir hata (BORDER ve / veya ÇATIŞMA) ortaya çıktığında , yerleştirme. Yerleşim başarılı olursa, işleci işleçlerin örnek değişkenine ekleyin OperatorGrid örneğinin. Bu değişken, işleçleri ekleme sıralarında içermelidir.

Yerleşim başarılı olursa, **place_operator** () yöntemi **true** değerini döndürür. Aksi takdirde, **yanlış** döndürür. İçinde Ayrıca, **place_operator** () yönteminin sonunda :

a) Operatör operatöre başarıyla yerleştirildiyse aşağıdaki mesajı yazdırın (bkz. <u>Şekil 2</u> ve **OperatorGrid** örneği:

BAŞARI: **Op_size** boyutunda operatör **işareti** yerleştirilir (**center_x** , **center_y**).

işareti operatörü, **center_x** ve **center_y** işaretidir nerede operatörün merkezi konuma vardır (üst sınıf **Operatöründe** tanımlanır) ve **op_size** , operatörün boyutudur (

üst sınıf **Operatör**).
b) **SINIR HATASI varsa** (bkz. <u>Şekil 4</u>), yerleşimi atlayın ve aşağıdakileri yazdırın mesaj:

SINIR HATASI: op_size boyutunda operatör işareti yerleştirilemez (center_x , center_y).

işareti operatörü, **center_x** ve **center_y** işaretidir nerede operatörün merkezi konuma vardır (üst sınıf **Operatöründe** tanımlanır) ve **op_size**, operatörün boyutudur (üst sınıf **Operatör**).

c) Bir ÇATIŞMA HATASI varsa (bkz. Şekil 5), yerleşimi atlayın ve aşağıdakileri yazdırın mesaj:

ÇATIŞMA HATASI: op_size boyutunda operatör işareti yerleştirilemez (center_x , center_y).

işareti operatörü, center_x ve center_y işaretidir nerede operatörün merkezi konuma vardır (üst sınıf Operatöründe tanımlanır) ve op_size, operatörün boyutudur (üst sınıf Operatör).

d) Hem SINIR hem de ÇATIŞMA HATASI varsa, yerleşimi atlayın ve yazdırın. sırayla aşağıdaki mesajları takip edin:

Sayfa 2

SINIR HATASI: op_size boyutunda operatör işareti yerleştirilemez (center_x , center_y). ÇATIŞMA HATASI: op_size boyutunda operatör işareti yerleştirilemez (center_x , center_y).

işareti operatörü, **center_x** ve **center_y** işaretidir nerede operatörün merkezi konuma vardır (üst sınıf **Operatöründe** tanımlanır) ve **op_size** , operatörün boyutudur (üst sınıf **Operatör**).

Yerleştirilen Operatörlere Yönelik Talimatlar

Move_operator (int x, int y, Char yönü, int move_by) yöntemi merkezini taşır Operatör verilen yöne belirtilen miktarda (sağ için 'R', sola için 'L', yukarı için 'U', aşağı için 'D'). Burada, x ve y, operatörün bir bölümünün bulunduğu ızgaradaki bir hücreyi belirtir. Hücre (x, y) Yapmaz Operatörün merkezi olmak ZORUNDA. Önce merkezi bulmalı ve operatörü Hareket miktarı (örneğin, bakınız Şekil 6).

	1	2	3	4	5	move_operator (1,2 'R", 2) VEYA		1	2	3	4	5
1		+				we_operator (2,1 'R", 2)	1				+	
2	+	+	+			+ İşlecinin merkezini taşı (2,2) ila (2,4) arasında	2			+	+	+
3		+					3				+	
4						move_operator (2,2 'R", 2) VEYA move_operator (2,3 'R", 2)	4					
5						VEYA move_operator (3,2 'R", 2)	5					

Şekil 6: move_operator () Doğru Yön Örneği

Taşıma herhangi bir sınır ve / veya çakışma hatasına neden olmazsa, move_operator () yöntemi geri döner doğru. Aksi takdirde, yanlış döndürür. Ayrıca, move_operator () yönteminin sonunda:

a) Operatör başarıyla taşınırsa aşağıdaki mesajı yazdırın:

BAŞARILI: (x_after, y_after) ila (x_before, y_before) taşınmış işareti.

işaretidir nerede operatörün işareti taşınmış **x_before** ve merkezi konumu **y_before** vardır **Taşımadan** önce operatör, ve **x_after** ve **y_after** operatörden sonraki operatörün merkezi konumudur. hareket.

b) **BORDER ERROR varsa** (bkz. <u>Şekil 4</u>), taşıma talimatını atlayın ve aşağıdaki mesaj:

SINIR HATASI: işareti (x_before , y_before) 'den (x_after , y_after)' a taşınamaz .

nerede **işareti** taşınamaz operatörün işareti, **x_before** ve merkez **y_before** vardır operatörün taşıma girişiminden önceki konumu ve **x_after** ve **y_after** Hedef hareketi için operatör.

c) Bir ÇATIŞMA HATASI varsa (bkz. <u>Şekil 5</u>), taşıma talimatını atlayın ve yazdırın. aşağıdaki mesaj:

ÇATIŞMA HATASI: işareti (x_before , y_before) 'den (x_after , y_after)' a taşınamaz .

Sayfa 3

nerede **işareti** taşınamaz operatörün işareti, **x_before** ve merkez **y_before** vardır operatörün taşıma girişiminden önceki konumu ve **x_after** ve **y_after** Hedef hareketi için operatör.

d) Hem **SINIR** hem de **ÇATIŞMA HATASI varsa**, taşıma talimatını atlayın ve yazdırın. **sırayla** aşağıdaki mesajları takip edin :

nerede **işareti** taşınamaz operatörün işareti, **x_before** ve merkez **y_before** vardır operatörün taşıma girişiminden önceki konumu ve **x_after** ve **y_after** Hedef hareketi için operatör.

print_operators () metodu operatörleri eklenir operatörler print_operator () yöntemini çağırdığında örnek değişkeni. Unutmayın ki, yalnızca başarılı bir şekilde yerleştirilmiş operatörler İşleçler örneği, yerleştirme sırasına göre değişken (örneğin önceden eklenmiş işleç daha düşük işleçler dizinde dizin). Bu yüzden, içine yerleştirildikleri sırada yazdırılmalıdırlar. işleçler örnek değişkeni.

Gönderme ve Kurallar

- 1. Kaynak kod dosyanızın ismi **atama2.cpp** olmalıdır .
- 2. Programınız bir Linux sisteminde aşağıdaki komutu kullanarak derlenecektir. Yapamazsa Bu komut kullanılarak derlenip linklenerek puanlandırılmayacaktır (başarısız gönderim).

 g ++ -Wall -Werror ataması2.cpp -o ataması2
- 3. Programınız kullanarak kontrol edilecektir <u>Calico</u> (<u>https://bitbucket.org/uyar/calico</u>) otomatik denetleyicisi. Bu nedenle, **mesajları tam olarak ödevde verilen şekilde yazdırdığınızdan** emin olun. **tanım.**
- 4. Verilen sınıf tanımlarına başka yardımcı yöntemler **eklemenize izin** verilir. Ancak, dikkatli olun **Test dosyalarının** bu **main** () metodu **sadece** inşa etmek için verilen sınıf tanımlarını kullanacaktır. **Operator** (ler), **ArithmeticOperator** (lar) ve **OperatorGrid** (in) örnekleri . Yöntem çağırır Oluşturulan örnekler yukarıdaki sınıf tanımlarına göre yapılacaktır. **Kodun olacak Yukarıda verilen sınıf tanımlarına uymadığınızda testlerde başarısız olun.**
- 5. Kodlama stilinizin uygun ve tutarlı olduğundan emin olun. Gerekirse **clang-format aracını** kullanın. Değişken isimlerini İngilizce'den başka bir dilde kullanmayın.
- 6. Kaynak kodunuzun anlaşılmasını kolaylaştırmak için yorumlar ekleyin.
- 7. Bu **bireysel bir** ödevdir. Herhangi bir biçimde işbirliğine izin verilmez. **Çalışmak yok birlikte**, sınıf arkadaşlarınıza verilen kodun gösterilmesi dahil olmak üzere hiçbir şekilde paylaşılan kod yok fikirler.
- 8. Gönderdiğiniz tüm kodlar kendi kodunuz olmalıdır. Herhangi bir kod parçasını kopyalamayın / yapıştırmayın İnternette bulduğunuz her şeyi içeren kaynak.
- 9. Ödevler, hem otomatik araçlar hem de el **kitapları** kullanılarak **intihal kontrolü için kontrol** edilecektir. denetim. İntihal ve / veya fikri mülkiyet ihlali içeren herhangi bir görev derecelendirilmeyecek ve daha fazla disiplin işlemine tabi olacaktır.