



C Piscine

C 11

*Summary: Bu döküman 42'deki C Piscine'sinin C 11 modülünün dersidir.*

*Version:*

# Contents

I	Yönergeler	2
II	Önsöz	4
III	Egzersiz 00 : ft_foreach	5
IV	Egzersiz 01 : ft_map	6
V	Egzersiz 02 : ft_any	7
VI	Egzersiz 03 : ft_count_if	8
VII	Egzersiz 04 : ft_is_sort	9
VIII	Egzersiz 05 : do-op	10
IX	Egzersiz 06 : ft_sort_string_tab	12
X	Egzersiz 07 : ft_advanced_sort_string_tab	13

# Chapter I

## Yönergeler

- Lütfen sadece bu sayfayı referans alınız: söylentilere kulak asmayınız.
- Dikkat! Dokümanın gönderim öncesinde değişme ihtimali vardır.
- Lütfen dosyalarınız ve dizileriniz için gerekli yetkilere sahip olduğunuzdan emin olunuz.
- Bütün çalışmalarınız için gönderim talimatlarını takip ediniz.
- Çalışmalarınız sınıf arkadaşlarınız tarafından kontrol edilip notlandırılacaktır.
- Aynı zamanda, çalışmalarınız Moulinette adlı program tarafından da kontrol edilip notlandırılacaktır.
- Moulinette değerlendirmelerinde çok titiz ve katıdır. Otomatik bir program olmasından dolayı görüş alışverişi mümkün değildir. Sürpriz bir sonuçla karşılaşmak için çalışmalarınızı dikkatlice yapınız.
- Moulinette çok açık görüşlü değildir. Kodunuz Norm'a uymadığı takdirde onu anlamaya çalışmayacaktır. Moulinette dosyalarınızın norm'a uyup uymadığını kontrol etmek için **norminette** adında bir program kullanmaktadır. TL;DR: **norminette**'in kontrolünden geçemeyecek bir dosya teslim etmek akılsızca olacaktır.
- Çalışmalar en kolaydan en zora olacak şekilde zorluklarına göre sıralanmıştır. Daha zor bir çalışma başarıyla tamamlanmış bile olsa daha kolay bir çalışmanın tamamıyla fonksiyonel olmaması durumunda dikkate **alınmayacaktır**.
- Yasaklanmış bir fonksiyon kullanmak hile olarak görülmektedir. Bunu yapan kişiler -42 puan alacaktır, ve bu not pazarlığa tabi değildir.
- Sizden program istersek sadece bir main() fonksiyonu göndermeniz gerekir.
- Moulinette çalışmaları şu şekilde sınıflandırır: -Wall -Wextra -Werror ve gcc
- Eğer programınız sınıflandırılmazsa, 0 alırsınız.
- Dizinizde konunun başlığındakiler dışında hiçbir dosya bırakmayınız.
- Bir sorunuz mu var? Sağınızdaki arkadaşınıza sorun. Olmadı solunuzdakine...

- Başvuru kılavuzunuzun adı `Google / man / the Internet / ...` 'dır.
- Intranetteki forumun "C Piscine" kısmını ya da Slack'deki Piscine bölümünü kontrol edin.
- Konu içerisinde net bir şekilde belirtilmemiş detayları anlayabilmek için örnekleri dikkatlice inceleyiniz.
- Odin ve Thor adına ! Kafayı çalıştırın !!!

# Chapter II

## Önsöz

Küçük bir hikaye:

(1982, Kaliforniya) Los Angeles'dan Larry Walters her yıl absürt şekillerde kendilerini gösterir. Bir gün "I'm staying on the ground. I've proved the thing works."

Meteoroloji balonu planını, dışarıda, "oldukça rahatsız" Sears katlanır sandalyesinde

Larry, çapayla olan bağlantıyı kopartıp, arka bahçesinde yaklaşık 9 metre yüksekliğe

Arkadaşları Larry'nin katlanır sandalyesini arabasına bağlı tutan halatı kestiklerinde

Yaklaşık 300 metrede ya da 3000 metrede düz uçuşa geçemedi. Uzun süre yükseldikten sonra

0 yükseklikte, dengesinin bozulacağı ve başının derde gireceğini hissettiğinden, balonu


Eninde sonunda, birkaç balonu vuracak cesareti kendinde buldu ve yavaşça inişe geçti.

Bu cevap Federal Havacılık İdaresi'nin hoşuna gitmemişti. Bir Güvenlik Müfettişi olan

Bu hikayeden çıkarılacak ders, Larry Walters'ın uçmak yerine sandalyesinde oturup C öğrenmesi gerektiğidir....

# Chapter III

## Egzersiz 00 : ft\_foreach

	Exercise 00
ft_foreach	
Turn-in directory : <i>ex00/</i>	
Files to turn in : <b>ft_foreach.c</b>	
Allowed functions : None	

- Verilen int dizisinin içindeki bütün sayılara verilen fonksiyonu uygulayan **ft\_foreach** adında bir fonksiyon tanımlayın. Uygulama sırası dizi ile aynı olmalıdır.
- Fonksiyonun prototipi şu şekilde olmalıdır :


```
void ft_foreach(int *tab, int length, void(*f)(int));
```

- Örneğin, **ft\_foreach** fonksiyonu dizinin içindeki bütün sayıları basmak için için aşağıdaki çağırılabilir :

```
ft_foreach(tab, 1337, &ft_putnbr);
```

# Chapter IV

## Egzersiz 01 : ft\_map


	Exercise 01
ft_map	
Turn-in directory : <i>ex01/</i>	
Files to turn in : <b>ft_map.c</b>	
Allowed functions : <b>malloc</b>	

- Verilmiş bir ints dizisi için, dizi sırasına göre her ögesine bir fonksiyon uygulayan ve çıkışında döndürdüğü değerlerden yeni bir dizi oluşturan bir **ft\_map** fonksiyonu oluşturun.
- Fonksiyon dizilimin sıralamasına göre uygulanacaktır.
- Fonksiyonun prototipi şu şekilde olmalıdır :

```
int      *ft_map(int *tab, int length, int(*f)(int));
```

# Chapter V

## Egzersiz 02 : ft\_any

	Exercise 02
ft_any	
Turn-in directory : <i>ex02/</i>	
Files to turn in : <b>ft_any.c</b>	
Allowed functions : <b>None</b>	

- **ft\_any** adında bir fonksiyon yazın. Verilen `char **` dizisinin içindeki bütün elemanları **f** fonksiyonuna verin. Eğer fonksiyon her hangi bir elemanda 0 dışında birşey döndürürse 1 yoksa 0 döndürün.
- Fonksiyon dizilimin sıralamasına göre uygulanacaktır.
- Fonksiyonun prototipi şu şekilde olmalıdır :


```
int ft_any(char **tab, int(*f)(char*));
```

- Dizilim bir boş işaretçisi ile sınırlandırılacaktır.



# Chapter VI

## Egzersiz 03 : ft\_count\_if


	Exercise 03
ft_count_if	
Turn-in directory : <i>ex03/</i>	
Files to turn in : <b>ft_count_if.c</b>	
Allowed functions : <b>None</b>	

- **ft\_any** adında bir fonksiyon yazın. Verilen `char **` dizisinin içindeki bütün elemanları **f** fonksiyonuna verin ve kaçtane elemanın 0 dışında birşey döndürdüğünü sayın.
- Fonksiyon dizilimin sıralamasına göre uygulanacaktır.
- Fonksiyonun prototipi şu şekilde olmalıdır :

```
int ft_count_if(char **tab, int length, int(*f)(char*));
```

# Chapter VII

## Egzersiz 04 : ft\_is\_sort


	Exercise 04
ft_is_sort	
Turn-in directory : <i>ex04/</i>	
Files to turn in : <b>ft_is_sort.c</b>	
Allowed functions : None	

- Eğer dizi sıralı ise 1 değil ise 0 döndüren **ft\_is\_sort** adında bir fonksiyon tanımlayın.
- Argüman olarak verilen fonksiyon, eğer ilk değişken ikinciden daha düşük bir değer ise negatif bir tamsayı, eşit ise 0, diğer her durum için pozitif bir tamsayı döndürür.
- Fonksiyonun prototipi şu şekilde olmalıdır :

```
int ft_is_sort(int *tab, int length, int(*f)(int, int));
```

# Chapter VIII

## Egzersiz 05 : do-op

	Exercise 05
do-op	
Turn-in directory : <i>ex05/</i>	
Files to turn in : <b>Your program files</b>	
Allowed functions : <b>write</b>	

- do-op isimli bir program oluşturun.
- Program üç argüman ile çalıştırılacaktır: do-op value1 operateur value2
- Örnek :

```
$>./do-op 42 "+" 21
63
$>
```

- operator'ı halletmek için bir fonksiyon işaret eden dizi kullanmalısınız.
- Geçersiz bir operasyon durumunda programınız 0 yazdırmalıdır.
- Argüman sayısı geçersiz ise do-op bir şey görüntüleyemez.
- Programınız şu operasyonları yapabilmeli ve yazdırabilmelidir: '+' '-' '/' '\*' and '%'
- 0 ile bölünme durumunda şu sonucu yazdırmalıdır:

```
Stop : division by zero
```

- 0 ile modülo işlemi durumunda şu sonucu yazdırmalıdır:


```
Stop : modulo by zero
```

- Aşağıda Moulinette'in yürüteceği bazı işlemler listelenmiştir :

```
$> make clean
$> make
$> ./do-op
$> ./do-op 1 + 1
2
$> ./do-op 42amis - ---20toto12
62
$> ./do-op 1 p 1
0
$> ./do-op 1 + toto3
1
$>
$> ./do-op toto3 + 4
4
$> ./do-op foo plus bar
0
$> ./do-op 25 / 0
Stop : division by zero
$> ./do-op 25 % 0
Stop : modulo by zero
$>
```

# Chapter IX

## Egzersiz 06 : ft\_sort\_string\_tab


	Exercise 06
ft_sort_string_tab	
Turn-in directory : <i>ex06/</i>	
Files to turn in : <b>ft_sort_string_tab.c</b>	
Allowed functions : None	

- `ft_sort_string_tab` fonksiyonunu `ascii`'ye göre `tab`'deki stringleri sıralayacak.
- `tab` null ile sonlandırılmıştır.
- Sıralama dizinin işaretçilerinin değiş tokuşu ile gerçekleştirilecektir.
- Fonksiyonun prototipi şu şekilde olmalıdır :

```
void ft_sort_string_tab(char **tab);
```

# Chapter X

## Egzersiz 07 : ft\_advanced\_sort\_string\_tab

	Exercise 07
ft_advanced_sort_string_tab	
Turn-in directory : <i>ex07/</i>	
Files to turn in : <b>ft_advanced_sort_string_tab.c</b>	
Allowed functions : None	

- Değişken olarak verilen fonksiyonun geri dönüşüne bağlı olarak sıralama yapan bir `ft_advanced_sort_string_tab` fonksiyonu oluşturun.
- Sıralama işaretleyicilerinin değiş tokuşu ile gerçekleştirilecektir.
- `tab` anlamsız olup sonlandırılacaktır
- Fonksiyonun prototipi şu şekilde olmalıdır :

```
void ft_advanced_sort_string_tab(char **tab, int(*cmp)(char *, char *));
```



Calling `ft_advanced_sort_string_tab()` ile beraber ikincil değişken olarak `with ft_strcmp` çağırmak `ft_sort_string_tab()` ile aynı sonuca sebep olur.