



C Piscine

C 05

Summary: Bu doküman C Piscine @ 42 içindeki C 05 modülünün konusudur.

Contents

I	Yönergeler	2
II	Önsöz	4
III	Çalışma 00 : ft_iterative_factorial	6
IV	Çalışma 01 : ft_recursive_factorial	7
V	Çalışma 02 : ft_iterative_power	8
VI	Çalışma 03 : ft_recursive_power	9
VII	Çalışma 04 : ft_fibonacci	10
VIII	Çalışma 05 : ft_sqrt	11
IX	Çalışma 06 : ft_is_prime	12
X	Çalışma 07 : ft_find_next_prime	13
XI	Çalışma 08 : The Ten Queens	14

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. Süpriz 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 / insan / 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 !!!



Norminette `-R CheckForbiddenSourceHeader` işareti ile başlatılmalıdır
. Moulinette de bunu kullanacaktır.

Chapter II

Önsöz

Harry Potter efsanesinden alıntılar:

Oh you may not think me pretty,
But don't judge on what you see,
I'll eat myself if you can find
A smarter hat than me.

You can keep your bowlers black,
Your top hats sleek and tall,
For I'm the Hogwarts Sorting Hat
And I can cap them all.

The Sorting Hat, stored in the Headmaster's Office.
There's nothing hidden in your head
The Sorting Hat can't see,
So try me on and I will tell you
Where you ought to be.

You might belong in Gryffindor,
Where dwell the brave at heart,
Their daring, nerve, and chivalry
Set Gryffindors apart;

You might belong in Hufflepuff,
Where they are just and loyal,
Those patient Hufflepuffs are true
And unafraid of toil;

Or yet in wise old Ravenclaw,
If you've a ready mind,
Where those of wit and learning,
Will always find their kind;

Or perhaps in Slytherin
You'll make your real friends,
Those cunning folks use any means

To achieve their ends.

So put me on! Don't be afraid!

And don't get in a flap!


You're in safe hands (though I have none)

For I'm a Thinking Cap!

Ne yazık ki, bu dersin Harry Potter efsanesi ile hiçbir alakası yoktur, çünkü çalışmalarınız sihir kullanarak tamamlanamayacak.

Chapter III

Çalışma 00 : ft_iterative_factorial


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

- Parametre olarak verilen sayının faktöryelini çıktı veren iteratif bir fonksiyon yazınız.
- Eğer argüman geçersizse fonksiyon 0 sonucunu vermelidir.
- Overflow durumları işlenmemelidir , fonksiyon çıktısı tanımsız olacaktır.
- Prototip şu şekilde olmalıdır :

```
int ft_iterative_factorial(int nb);
```

Chapter IV

Çalışma 01 : ft_recursive_factorial


	Exercise 01
ft_recursive_factorial	
Turn-in directory : <i>ex01/</i>	
Files to turn in : ft_recursive_factorial.c	
Allowed functions : None	

- Parametre olarak verilen sayının faktöryelini çıktı veren recursive bir fonksiyon yazınız.
- Eğer argüman geçersizse fonksiyon 0 sonucunu vermelidir.
- Overflow durumları işlenmemelidir , fonksiyon çıktısı tanımsız olacaktır.
- Prototip şu şekilde olmalıdır :

```
int ft_recursive_factorial(int nb);
```


Chapter V

Çalışma 02 : ft_iterative_power


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

- Parametre olarak bir sayının kuvvetini çıktı veren iteratif bir fonksiyon oluşturunuz. 0'dan küçük bir kuvvet 0 çıktısını verir. Overflow durumları işlenmemelidir.
- 0'ın 0'ıncı kuvvetinin 1 çıktısını vereceğine karar verdik
- Prototip şu şekilde olmalıdır :

```
int ft_iterative_power(int nb, int power);
```

Chapter VI

Çalışma 03 : ft_recursive_power


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

- Parametre olarak bir sayının kuvvetini çıktı veren recursive bir fonksiyon oluşturunuz.
- Overflow durumları işlenmemelidir, çıktı tanımsız olur.
- 0'ın 0'ıncı kuvvetinin 1 çıktısını vereceğine karar verdik
- Prototip şu şekilde olmalıdır :

```
int ft_recursive_power(int nb, int power);
```

Chapter VII

Çalışma 04 : ft_fibonacci

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


- İlk ögesi 0 indeksinde olmak üzere Fibonacci serisinin **n**-inci ögesini çıktı veren bir **ft_fibonacci** fonksiyonu oluşturunuz. Fibonacci serisinin şu şekilde başladığını göz önünde bulunduracağız: 0, 1, 1, 2.
- Overflow durumları işlenmemelidir, çıktı tanımsız olur.
- Prototip şu şekilde olmalıdır :

```
int ft_fibonacci(int index);
```

- Tabii ki de, **ft_fibonacci** recursive olmalıdır.
- Eğer **index** 0'dan küçükse, fonksiyon -1 sonucunu vermelidir.

Chapter VIII

Çalışma 05 : ft_sqrt


	Exercise 05
	ft_sqrt
	Turn-in directory : <i>ex05/</i>
	Files to turn in : ft_sqrt.c
	Allowed functions : None

- Bir sayının karekökünü çıktı olarak veren bir fonksiyon oluşturunuz (eğer karekökü varsa), sayının karekökü irrasyonelse fonksiyon 0 sonucunu versin.
- Prototip şu şekilde olmalıdır :

```
int ft_sqrt(int nb);
```

Chapter IX

Çalışma 06 : ft_is_prime

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

- Parametre olarak verilen sayı asal sayı ise 1, asal sayı değil ise 0 sonucunu veren bir fonksiyon oluşturunuz.
- Prototip şu şekilde olmalıdır :


```
int ft_is_prime(int nb);
```



0 ve 1 asal sayı değildirler.

Chapter X

Çalışma 07 : ft_find_next_prime


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

- Değişken olarak verilen sayıdan büyük ya da eşit en küçük asal sayıyı çıktı veren bir fonksiyon oluşturunuz.
- Prototip şu şekilde olmalıdır :

```
int ft_find_next_prime(int nb);
```

Chapter XI

Çalışma 08 : The Ten Queens

	Exercise 08
	The Ten Queens
	Turn-in directory : <i>ex08/</i>
	Files to turn in : <i>ft_ten_queens_puzzle.c</i>
	Allowed functions : <i>write</i>

- Ona onluk bir satranç tahtasında on tane vezirin birbirine tek hamlede ulaşamayacak şekildeki dizimlerini gösteren ve bu dizimlerin sayısını çıktı veren bir fonksiyon oluşturunuz.
- Bunu çözmek için recursive bir fonksiyon gerekmektedir.
- Prototip şu şekilde olmalıdır :

```
int ft_ten_queens_puzzle(void);
```

- Şu şekilde gösterilmelidir :

```
$>./a.out | cat -e
0257948136$
0258693147$
...
4605713829$
4609582731$
...
9742051863$
$>
```

- Sıra soldan sağa gitmektedir. İlk rakam ilk Vezir'in ilk sütündeki yerini temsil etmektedir (indeks 0'dan başlayan şekilde). N'inci rakam N'inci Vezir'in N'inci sütündeki yerini temsil etmektedir.
- Sonuç değeri gösterilen çözümlerin toplam sayısına eşit olmalıdır.