

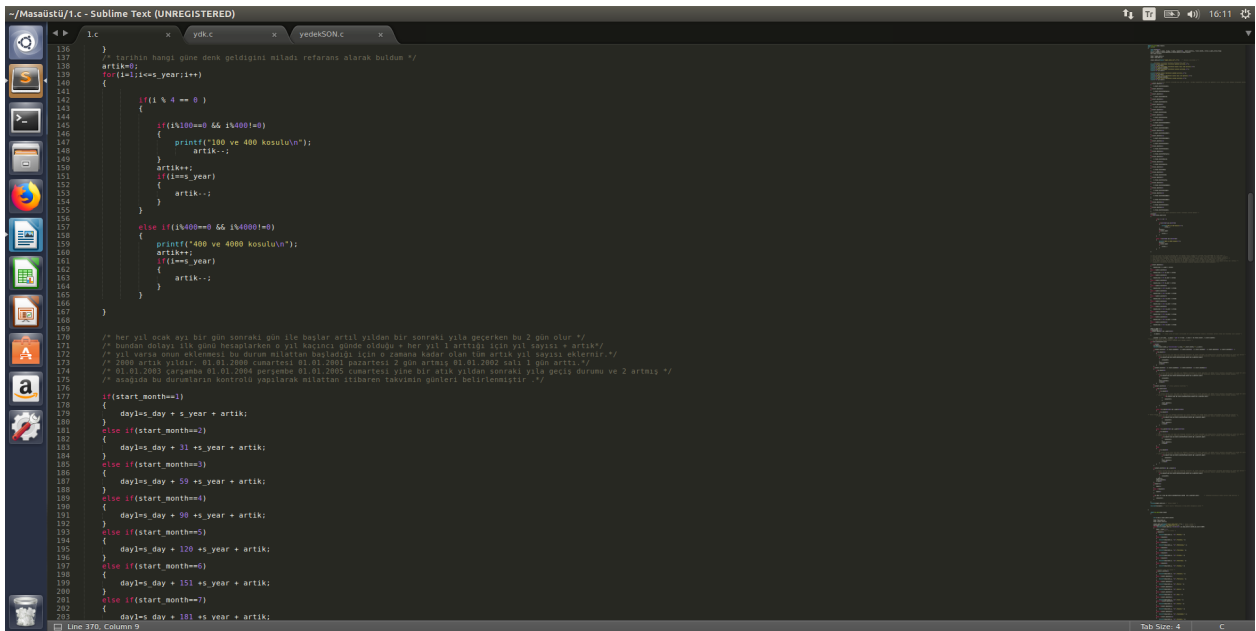
HW7 PART2 AÇIKLAMA

Typedef enum month ile ayların son gününün yılın kaçınıcı gününe denk geldiğini yazdım. Böylelikle başlangıç gününü bulurken ayın kaçınıcı günü ve bir sonraki ayın yılın kaçınıcı günü olduğunu bilerek başlangıç tarihinin yılın kaçınıcı günü olduğunu basitçe öğrenmiş oluyorum.

Typedef enum day ile new_date textte sıralı bi şekilde günleri yazmayı sağladım.

FILE ile dosyalarım için pointer belirledikten sonra input date dosyası açtım.

Kullanıcıdan başlangıç tarihi ve bitiş tarihini aldıktan sonra yukarda belirttiğim typedef enum month ve if koşulları aracılığı ile başlangıç ve bitiş tarihleri arasında kaç gün olacağını bulmaya başladım. Bu durumu daha net hale sokmak için artık yıl hesapları yapılması gerekirdi.



```
136 }
137
138 /* tarihini hangi güne denk geldiğini buladı referans olarak buldu */
139 artik++;
140 for(i=1; i==s_year; i++)
141 {
142     if(i % 4 == 0)
143     {
144         if(i % 100 == 0 && i % 400 != 0)
145         {
146             printf("100 ve 400 kuralına");
147             artik--;
148         }
149         artik++;
150         if(i==s_year)
151         {
152             artik--;
153         }
154     }
155     else if(i % 400 == 0 && i % 400 != 0)
156     {
157         printf("400 ve 4000 kuralına");
158         artik++;
159         if(i==s_year)
160         {
161             artik--;
162         }
163     }
164 }
165
166
167
168
169
170 /* her yıl ocak ayı bir önceki yılın 1 ocak ayından bir sonraki gün ile başlar .Artık yıldan bir sonraki
171 yıla geçerken bu 2 gün olur bundan dolayı ilk günü hesaplariken o yıl kaçınıcı günde olduğu + her yıl 1
172 arttığı için yıl sayısı + artık yıl varsa onun eklenmesi bu durum milattan başladığı için o zamana
173 kadar olan tüm artık yıl sayısı eklernir. 2000 artık yıldır. 01.01.2000 cumartesi 01.01.2001 pazartesi 2 gün
174 artmış 01.01.2002 salı 1 gün arttı. 01.01.2003 çarşamba 01.01.2004 perşembe 01.01.2005 cumartesi yine bir
175 atık yıldan sonraki yıla geçiş durumu ve 2 artmış */
176
177 /* milattan bu günümüze kontrolü yapılarak milattan itibaren takvimin günleri belirlenmiştir */
178
179 if(start_month==1)
180 {
181     day=s_day + s_year + artik;
182 }
183 else if(start_month==2)
184 {
185     day=s_day + 31 + s_year + artik;
186 }
187 else if(start_month==3)
188 {
189     day=s_day + 59 + s_year + artik;
190 }
191 else if(start_month==4)
192 {
193     day=s_day + 90 + s_year + artik;
194 }
195 else if(start_month==5)
196 {
197     day=s_day + 120 + s_year + artik;
198 }
199 else if(start_month==6)
200 {
201     day=s_day + 151 + s_year + artik;
202 }
203 else if(start_month==7)
204 {
205     day=s_day + 181 + s_year + artik;
206 }
207 else if(start_month==8)
208 {
209     day=s_day + 212 + s_year + artik;
210 }
211 else if(start_month==9)
212 {
213     day=s_day + 243 + s_year + artik;
214 }
215 else if(start_month==10)
216 {
217     day=s_day + 273 + s_year + artik;
218 }
219 else if(start_month==11)
220 {
221     day=s_day + 304 + s_year + artik;
222 }
223 else if(start_month==12)
224 {
225     day=s_day + 334 + s_year + artik;
226 }
227
228 return day;
229 }
```

Bu koşulların altında milattan başlayarak 4 yıl da bir artık yıl olur ancak yıl 100 tam bölünüp 400 tam bölünmüyorsa artık yıldır bu koşulun dışında artık yıl değildir. Bu durumda artık yıl hesabı 2 defa yapılması diye artık count u ile 4 yıl içine girip bu durumda girerse artık tan 1 eksilttim. Başlangıç tarihinin yılını tutan s_year artık yıl ise 1 çıkardım çünkü 29 şubat öncesi durumlarda yanlış bilgi gösterirdi. (Artık yılın artık günü 29 şubatta dahil olduğundan ocak 1 1 arttırıldı örneğin pazartesi ise salı yazardı.) 2000 li yıllarda olmamıza rağmen yıl 400 e bölünüp 4000 e bölünmeyen durumlarda da artık yıl olma koşulu vardır. bunları da 4 yılda bir durumdan çıkarmak gerekirken fazladan artık gün eklenmesin. Bu durumda programa yazarak takvimin aralığını genişletmiş oldum.

Her yıl ocak ayı bir önceki yılın 1 ocak ayından bir sonraki gün ile başlar .Artık yıldan bir sonraki yıla geçerken bu 2 gün olur bundan dolayı ilk günü hesaplariken o yıl kaçınıcı günde olduğu + her yıl 1 arttığı için yıl sayısı + artık yıl varsa onun eklenmesi bu durum milattan başladığı için o zamana kadar olan tüm artık yıl sayısı eklernir. 2000 artık yıldır. 01.01.2000 cumartesi 01.01.2001 pazartesi 2 gün artmış 01.01.2002 salı 1 gün arttı. 01.01.2003 çarşamba 01.01.2004 perşembe 01.01.2005 cumartesi yine bir atık yıldan sonraki yıla geçiş durumu ve 2 artmış.

Bu durumdan yola çıkarak typedef enum month değerlerini kullanarak 1 ocaktan sonra kaç gün gecti ise bunun mod 7 si ile aralarında fark oluşturduğu bilgisi ile başlangıç gününü buldum.

Ve bunu void new_date fonksiyonunda kullanmak için d_day değişkenine atadım böylelikle başlangıç günün hangi gün olduğunu artık biliyorum.

Ancak bu durum daha önceden bahsettiğim 29 şubat durumundan dolayı artık yıl harici yıllarda geçerli.Yıl artık yıl ise 29 şubattan sonraki bir tarih başlangıç durumunda ise d_day i 1 arttırıyorum ki (artık yılın başlangıç günü 29 şubat olarak düşünüyorum.) başangıç günü doğru bir şekilde bulunsun.

```
~/Masaüstü/1.c - Sublime Text (UNREGISTERED)
x ydk.c x yedesSON.c
295 {
296     i=counter;
297 }
298 start_month++;
299 s_day=0;
300 }
301 }
302
303 else if(s_year%400==0 && s_year%400!=0)
304 {
305     if(s_day==29)
306     {
307         /* bitiş tarihi ayın son günü ise döngüyü sonlandır bu işlem yapılmasa ise döngü bitiş tarihini geciktirir bu işlem for yerine */
308         /* while ile çok hızlıdır kullanılabilir ama burada ay ayda girerek programın hızını yavaş oranda tutmak istedin. */
309         if(s_day==f_day && start_month==final_month && s_year==f_year)
310         {
311             i=counter;
312             start_month++;
313             s_day=0;
314         }
315     }
316 }
317
318 else if(s_day==28)
319 {
320     /* bitiş tarihi ayın son günü ise döngüyü sonlandır bu işlem yapılmasa ise döngü bitiş tarihini geciktirir bu işlem for yerine */
321     /* while ile çok hızlıdır kullanılabilir ama burada ay ayda girerek programın hızını yavaş oranda tutmak istedin. */
322     if(s_day==f_day && start_month==final_month && s_year==f_year)
323     {
324         i=counter;
325         start_month++;
326         s_day=0;
327     }
328 }
329 }
330 }
331
332 if(start_month==12 && s_day==31)
333 {
334     /* bitiş tarihi ayın son günü ise döngüyü sonlandır bu işlem yapılmasa ise döngü bitiş tarihini geciktirir bu işlem for yerine */
335     /* while ile çok hızlıdır kullanılabilir ama burada ay ayda girerek programın hızını yavaş oranda tutmak istedin. */
336     if(s_day==f_day && start_month==final_month && s_year==f_year)
337     {
338         i=counter;
339         s_year++;
340         start_month=1;
341         s_day=0;
342     }
343 }
344
345 if(day!=0)
346 {
347     day++;
348 }
349 else if(day==0)
350 {
351     day=0;
352 }
353
354 if(s_day == f_day && start_month==final_month && s_year==f_year) /* döngünün bitiminin bitiş tarihi olan kontrol */
355 {
356     i=counter;
357 }
358
359 fclose(input_date_p); /* dosya kapat */
360
361 new_date(d_day); /* tarih çeviri fonksiyonu ve new_date dosyasına yazma */
362
363 }
364
365 void new_date(day day1)
366 {
367     int s_day,s_year,start_month;
368     FILE *new_date_p;
369     FILE *input_date_p;
370     input_date_p=fopen("input_date.txt","r"); /* okuma işlemi */
371     new_date_p=fopen("new_date.txt","w"); /* yazma işlemi */
372     while(fscanf(input_date_p,"%d/%d/%d",&s_day,&s_start_month,&s_year)==EOF)
373     {
374         day1 = day1 % 7;
375         /* döngüyü yeni ile yazma */
376         if(day1==1)
377         {
378             fprintf(new_date_p, "%s","Monday,");
379         }
380         else if(day1==2)
381         {
382             fprintf(new_date_p, "%s","Tuesday,");
383         }
384         else if(day1==3)
385         {
386             fprintf(new_date_p, "%s","Wednesday,");
387         }
388         else if(day1==4)
389         {
390             fprintf(new_date_p, "%s","Thursday,");
391         }
392         else if(day1==5)
393         {
394             fprintf(new_date_p, "%s","Friday,");
395         }
396         else if(day1==6)
397         {
398             fprintf(new_date_p, "%s","Saturday,");
399         }
400         else if(day1==0)
401         {
402             fprintf(new_date_p, "%s","Sunday,");
403         }
404     }
405     /* print date ile yazma */
406     if(start_month==1)
407     {
408         fprintf(new_date_p, "%s","January,");
409     }
410     else if(start_month==2)
411     {
412         fprintf(new_date_p, "%s","February,");
413     }
414     else if(start_month==3)
415     {
416         fprintf(new_date_p, "%s","March,");
417     }
418     else if(start_month==4)
419     {
420         fprintf(new_date_p, "%s","April,");
421     }
422     else if(start_month==5)
423     {
424         fprintf(new_date_p, "%s","May,");
425     }
426     else if(start_month==6)
427     {
428         fprintf(new_date_p, "%s","June,");
429     }
430     else if(start_month==7)
431     {
432         fprintf(new_date_p, "%s","July,");
433     }
434     else if(start_month==8)
435     {
436         fprintf(new_date_p, "%s","August,");
437     }
438     else if(start_month==9)
439     {
440         fprintf(new_date_p, "%s","September,");
441     }
442     else if(start_month==10)
443     {
444         fprintf(new_date_p, "%s","October,");
445     }
446     else if(start_month==11)
447     {
448         fprintf(new_date_p, "%s","November,");
449     }
450     else if(start_month==12)
451     {
452         fprintf(new_date_p, "%s","December,");
453     }
454     fprintf(new_date_p, "\n");
455     fclose(new_date_p);
456     fclose(input_date_p);
457 }
```

Hemen altında for ile daha önceden bulduğum başlangıç bitiş tarihi arasında kaç gün olduğu bilgisi ile tarihleri sıralamaya.Döngünün başında bu tarihleri fprintf ile input_date dosyasına atıyorum.

Döngü içinde artık yıldan kaynaklı 29 şubat durumları ve ay sonu geldiğinde s_day 0 layıp start_month değerini arttırıyorum.Ve tabiki de bu döngü içinde pazar olduğunda pazartesiye bağlayıp diğer durumlarda günleri 1 arttırıyorum.En sonunda artık yıl 29 şubat durumunda dolayı meydana gelebilecek döngü sonlanmasını garantiye almak için değişkenler bitiş tarihine eşit olduğunda döngüyü sonlandırıyorum.

Ve dosyayı kapatıp new_date fonksiyonunu çağırıyorum.

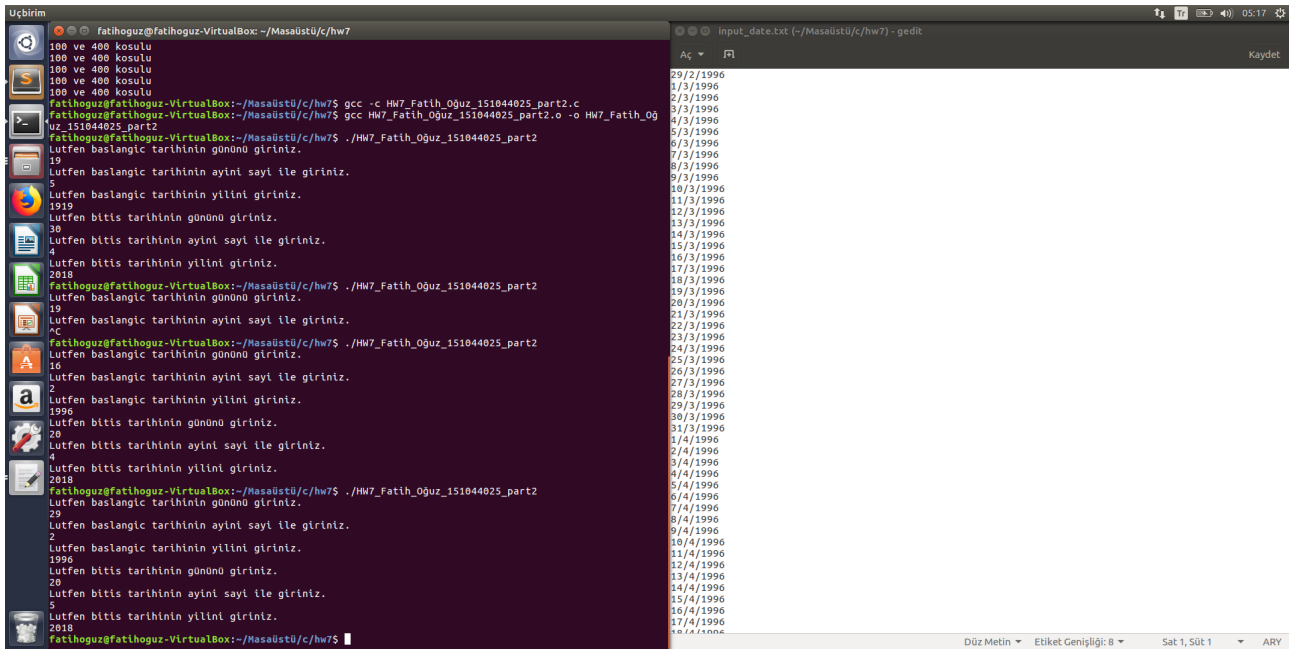
```
~/Masaüstü/1.c - Sublime Text (UNREGISTERED)
x ydk.c x yedesSON.c
358 }
359 fclose(input_date_p); /* dosya kapat */
360
361 new_date(d_day); /* tarih çeviri fonksiyonu ve new_date dosyasına yazma */
362
363 }
364
365 void new_date(day day1)
366 {
367     int s_day,s_year,start_month;
368     FILE *new_date_p;
369     FILE *input_date_p;
370     input_date_p=fopen("input_date.txt","r"); /* okuma işlemi */
371     new_date_p=fopen("new_date.txt","w"); /* yazma işlemi */
372     while(fscanf(input_date_p,"%d/%d/%d",&s_day,&s_start_month,&s_year)==EOF)
373     {
374         day1 = day1 % 7;
375         /* döngüyü yeni ile yazma */
376         if(day1==1)
377         {
378             fprintf(new_date_p, "%s","Monday,");
379         }
380         else if(day1==2)
381         {
382             fprintf(new_date_p, "%s","Tuesday,");
383         }
384         else if(day1==3)
385         {
386             fprintf(new_date_p, "%s","Wednesday,");
387         }
388         else if(day1==4)
389         {
390             fprintf(new_date_p, "%s","Thursday,");
391         }
392         else if(day1==5)
393         {
394             fprintf(new_date_p, "%s","Friday,");
395         }
396         else if(day1==6)
397         {
398             fprintf(new_date_p, "%s","Saturday,");
399         }
400         else if(day1==0)
401         {
402             fprintf(new_date_p, "%s","Sunday,");
403         }
404     }
405     /* print date ile yazma */
406     if(start_month==1)
407     {
408         fprintf(new_date_p, "%s","January,");
409     }
410     else if(start_month==2)
411     {
412         fprintf(new_date_p, "%s","February,");
413     }
414     else if(start_month==3)
415     {
416         fprintf(new_date_p, "%s","March,");
417     }
418     else if(start_month==4)
419     {
420         fprintf(new_date_p, "%s","April,");
421     }
422     else if(start_month==5)
423     {
424         fprintf(new_date_p, "%s","May,");
425     }
426     else if(start_month==6)
427     {
428         fprintf(new_date_p, "%s","June,");
429     }
430     else if(start_month==7)
431     {
432         fprintf(new_date_p, "%s","July,");
433     }
434     else if(start_month==8)
435     {
436         fprintf(new_date_p, "%s","August,");
437     }
438     else if(start_month==9)
439     {
440         fprintf(new_date_p, "%s","September,");
441     }
442     else if(start_month==10)
443     {
444         fprintf(new_date_p, "%s","October,");
445     }
446     else if(start_month==11)
447     {
448         fprintf(new_date_p, "%s","November,");
449     }
450     else if(start_month==12)
451     {
452         fprintf(new_date_p, "%s","December,");
453     }
454     fprintf(new_date_p, "\n");
455     fclose(new_date_p);
456     fclose(input_date_p);
457 }
```

Bu fonksiyonda new_date ve input_date dosyalarını açıyorum.
input_date dosyasından aldığım verileri kullanarak günleri ve ayları if koşulları altında yazıya döküp new_date dosyası formunda yazılması gereken hale getirirerek bunları new_date dosyasına atıyorum. Ve günü 1 artırarak input_date ten bir sonraki gün verisini çekiyorum .input_date dosyası EOF ' ı görene kadar.

Sonrasında dosyalarımı kapatıyorum.

```
~/Masüstü/1.c - Sublime Text (UNREGISTERED)
x y8kC y8kSON.c
400 fprintf(new_date_p, "%s", "Friday,");
401 }
402 else if(day!=0)
403 {
404     fprintf(new_date_p, "%s", "Saturday,");
405 }
406 else if(day!=0)
407 {
408     fprintf(new_date_p, "%s", "Sunday,");
409 }
410
411 /* ayları yazı ile yazma */
412 if(start_month==1)
413 {
414     fprintf(new_date_p, "%s", "January,");
415 }
416 else if(start_month==2)
417 {
418     fprintf(new_date_p, "%s", "February,");
419 }
420 else if(start_month==3)
421 {
422     fprintf(new_date_p, "%s", "March,");
423 }
424 else if(start_month==4)
425 {
426     fprintf(new_date_p, "%s", "April,");
427 }
428 else if(start_month==5)
429 {
430     fprintf(new_date_p, "%s", "May,");
431 }
432 else if(start_month==6)
433 {
434     fprintf(new_date_p, "%s", "June,");
435 }
436 else if(start_month==7)
437 {
438     fprintf(new_date_p, "%s", "July,");
439 }
440 else if(start_month==8)
441 {
442     fprintf(new_date_p, "%s", "August,");
443 }
444 else if(start_month==9)
445 {
446     fprintf(new_date_p, "%s", "September,");
447 }
448 else if(start_month==10)
449 {
450     fprintf(new_date_p, "%s", "October,");
451 }
452 else if(start_month==11)
453 {
454     fprintf(new_date_p, "%s", "November,");
455 }
456 else if(start_month==12)
457 {
458     fprintf(new_date_p, "%s", "December,");
459 }
460
461 /* new_date dosyasına yeni tarih veriyimene yazma */
462 fprintf(new_date_p, "%d,%d\n", s_day, s_year);
463 day++;
464 }
465 fclose(input_date_p);
466 fclose(new_date_p);
467 }
```

SCREENSHOTS



Metin Düzenleyici		new_date.txt (~/Masaüstü/c/hw7) - gedit		input_date.txt (~/Masaüstü/c/hw7) - gedit	
Aç		Kaydet		Aç	
Monday, May, 19, 1919				19/5/1919	
Tuesday, May, 20, 1919				20/5/1919	
Wednesday, May, 21, 1919				21/5/1919	
Thursday, May, 22, 1919				22/5/1919	
Friday, May, 23, 1919				23/5/1919	
Saturday, May, 24, 1919				24/5/1919	
Sunday, May, 25, 1919				25/5/1919	
Monday, May, 26, 1919				26/5/1919	
Tuesday, May, 27, 1919				27/5/1919	
Wednesday, May, 28, 1919				28/5/1919	
Thursday, May, 29, 1919				29/5/1919	
Friday, May, 30, 1919				30/5/1919	
Saturday, May, 31, 1919				31/5/1919	
Sunday, June, 1, 1919				1/6/1919	
Monday, June, 2, 1919				2/6/1919	
Tuesday, June, 3, 1919				3/6/1919	
Wednesday, June, 4, 1919				4/6/1919	
Thursday, June, 5, 1919				5/6/1919	
Friday, June, 6, 1919				6/6/1919	
Saturday, June, 7, 1919				7/6/1919	
Sunday, June, 8, 1919				8/6/1919	
Monday, June, 9, 1919				9/6/1919	
Tuesday, June, 10, 1919				10/6/1919	
Wednesday, June, 11, 1919				11/6/1919	
Thursday, June, 12, 1919				12/6/1919	
Friday, June, 13, 1919				13/6/1919	
Saturday, June, 14, 1919				14/6/1919	
Sunday, June, 15, 1919				15/6/1919	
Monday, June, 16, 1919				16/6/1919	
Tuesday, June, 17, 1919				17/6/1919	
Wednesday, June, 18, 1919				18/6/1919	
Thursday, June, 19, 1919				19/6/1919	
Friday, June, 20, 1919				20/6/1919	
Saturday, June, 21, 1919				21/6/1919	
Sunday, June, 22, 1919				22/6/1919	
Monday, June, 23, 1919				23/6/1919	
Tuesday, June, 24, 1919				24/6/1919	
Wednesday, June, 25, 1919				25/6/1919	
Thursday, June, 26, 1919				26/6/1919	
Friday, June, 27, 1919				27/6/1919	
Saturday, June, 28, 1919				28/6/1919	
Sunday, June, 29, 1919				29/6/1919	
Monday, June, 30, 1919				30/6/1919	
Tuesday, July, 1, 1919				1/7/1919	
Wednesday, July, 2, 1919				2/7/1919	
Thursday, July, 3, 1919				3/7/1919	
Friday, July, 4, 1919				4/7/1919	
Saturday, July, 5, 1919				5/7/1919	
Sunday, July, 6, 1919				6/7/1919	
Monday, July, 7, 1919				7/7/1919	
Düz Metin		Etiket Genişliği: 8		Düz Metin	
Sat 23, Süt 21		Sat 12, Süt 10		ARY	

Metin Düzenleyici		new_date.txt (~/Masaüstü/c/hw7) - gedit		input_date.txt (~/Masaüstü/c/hw7) - gedit	
Aç		Kaydet		Aç	
Friday, February, 16, 1996				16/2/1996	
Saturday, February, 17, 1996				17/2/1996	
Sunday, February, 18, 1996				18/2/1996	
Monday, February, 19, 1996				19/2/1996	
Tuesday, February, 20, 1996				20/2/1996	
Wednesday, February, 21, 1996				21/2/1996	
Thursday, February, 22, 1996				22/2/1996	
Friday, February, 23, 1996				23/2/1996	
Saturday, February, 24, 1996				24/2/1996	
Sunday, February, 25, 1996				25/2/1996	
Monday, February, 26, 1996				26/2/1996	
Tuesday, February, 27, 1996				27/2/1996	
Wednesday, February, 28, 1996				28/2/1996	
Thursday, February, 29, 1996				29/2/1996	
Friday, March, 1, 1996				1/3/1996	
Saturday, March, 2, 1996				2/3/1996	
Sunday, March, 3, 1996				3/3/1996	
Monday, March, 4, 1996				4/3/1996	
Tuesday, March, 5, 1996				5/3/1996	
Wednesday, March, 6, 1996				6/3/1996	
Thursday, March, 7, 1996				7/3/1996	
Friday, March, 8, 1996				8/3/1996	
Saturday, March, 9, 1996				9/3/1996	
Sunday, March, 10, 1996				10/3/1996	
Monday, March, 11, 1996				11/3/1996	
Tuesday, March, 12, 1996				12/3/1996	
Wednesday, March, 13, 1996				13/3/1996	
Thursday, March, 14, 1996				14/3/1996	
Friday, March, 15, 1996				15/3/1996	
Saturday, March, 16, 1996				16/3/1996	
Sunday, March, 17, 1996				17/3/1996	
Monday, March, 18, 1996				18/3/1996	
Tuesday, March, 19, 1996				19/3/1996	
Wednesday, March, 20, 1996				20/3/1996	
Thursday, March, 21, 1996				21/3/1996	
Friday, March, 22, 1996				22/3/1996	
Saturday, March, 23, 1996				23/3/1996	
Sunday, March, 24, 1996				24/3/1996	
Monday, March, 25, 1996				25/3/1996	
Tuesday, March, 26, 1996				26/3/1996	
Wednesday, March, 27, 1996				27/3/1996	
Thursday, March, 28, 1996				28/3/1996	
Friday, March, 29, 1996				29/3/1996	
Saturday, March, 30, 1996				30/3/1996	
Sunday, March, 31, 1996				31/3/1996	
Monday, April, 1, 1996				1/4/1996	
Tuesday, April, 2, 1996				2/4/1996	
Wednesday, April, 3, 1996				3/4/1996	
Thursday, April, 4, 1996				4/4/1996	
Friday, April, 5, 1996				5/4/1996	
Düz Metin		Etiket Genişliği: 8		Düz Metin	
Sat 1, Süt 1		Sat 1, Süt 1		ARY	

