

```
C:\D:\Tarih Hesaplama\bin\Debug\Tarih Hesaplama.exe
*** Menu *** (Bugun: 21.11.2007)

1. Verilen iki tarih arasinda kac gun fark oldugunun bulunmasi
2. Verilen bir tarihten bugune kadar gecen gun sayisinin bulunmasi
3. Verilen bir yil icerisinde verilen iki tarih arasinda gecen gun sayisinin bul
4. Verilen bir tarihten verilen gun sayisi kadar sonraki tarihin bulunmasi
5. Bugunden verilen gun sayisi kadar onceki tarihin bulunmasi
6. Bugunden verilen gun sayisi kadar sonraki tarihin bulunmasi
7. Cikis

Lutfen istediginiz islemin numarasini giriniz.
```

Tarih Hesaplama Yazılımı

Umut BENZER

05-06-7670

<http://www.ubenzer.com>

Ege Üniversitesi Bilgisayar Mühendisliği

Algoritma ve Programlama

Proje-1

Teslim Tarihi: 23.11.2007



Analiz

Program tarih hesaplamalarına dair çözümler sunar. İki tarih arası geçen gün sayısının bulunması, belirli bir tarihten belirli bir gün sonrasının ne zaman olacağı ve buna benzer soruları cevaplandırmak için bu yazılım kullanılabilir.

Program ilk olarak günümüzün tarihini alacak ve ardından da yapabileceği işlemlerin bir listesini gösterecektir. Bunlar aşağıda listelenmiştir:

1. İki tarih arasındaki gün farkını bulmak
2. Kullanıcıdan istenecek bir tarih ile bugün arasındaki gün sayısını bulmak
3. Aynı yıl içerisindeki iki tarih arasındaki gün sayısını bulmak (yıl kullanıcıdan istenecektir)
4. Kullanıcıdan istenecek bir tarihten belirli bir gün sonrasının tarihinin bulunması
5. Günümüzden belirli bir gün öncesinin tarihinin bulunması
6. Günümüzden belirli bir gün sonrasının tarihinin bulunması

Bunun dışında programı kapatmak için bir çıkış seçeneği olacaktır. Program hatalı girdilere hazırlıklı olmalı ve artık yılları da dikkate almalıdır. Program her işlem sonrası menüye geri dönmelidir.

Tasarım

Programın öncelikli sorunlarından biri kullanıcıdan hatalı veri almamaktır. Dışarıdan gelen verilerin doğruluğu kontrol edilmeden daha fazlasını istemek hatalı bir davranış olacaktır. Kullanıcıdan birçok değişik yerde veri girdisi isteneceğinden kontrol yapılarını fonksiyon olarak atamak çok mantıklı olacaktır.

Programın genel olarak iki tür çıktı verdiği görülür. Birincisi “iki tarih arası gün” tipi çıktıdır ve gün tipindedir. İkincisi ise “şu tarihten şu kadar gün önce/sonra” tipindedir ve tarih tipindedir. Problemleri bu şekilde ayırdıktan sonra program tasarlanmaya başlanabilir.

Programda aynı işi yapacak birden fazla kod olması gereksiz ve hatalıdır. Bu yüzden tarih büyüklüğü karşılaştırma, artık yıl olup olmadığını öğrenme gibi sık kullanılacak işlemler fonksiyon olarak tanımlanacaktır.

Tasarımla ilgili daha ayrıntılı bilgiler Programcı Kataloğu kısmında bulunabilir.

Programcı Kataloğu

Programda 11 fonksiyon (birisi main olmak üzere) bulunmaktadır. Programcı kataloğu, programda geçen tüm fonksiyonların ne yaptığını açıklamakta ve girdi-çıkı değerlerini anlatmaktadır. Fonksiyonlar alfabetik sıralanmıştır. Açıklamalardaki girdiler prototipteki sıraya göre açıklanmıştır. Bu katalog programın 21.11.2007 tarihinde tamamlanmış sürümüne aittir ve buradaki kaynak kodları ile güncel kaynak kodları arasında çok ufak değişiklikler olması mümkündür. Kaynak kodunun en güncel sürümünü ve fonksiyon fonksiyon parçalanmamış çevrimiçi kaynaklarda bulabilirsiniz: <http://www.ubenzer.com/proje1.c>

int ay_gecerli_mi(int);

Girdi: ay
Çıktı: -1,0

Integer tipindeki girdinin (ay) geçerli bir ay olup olmadığını denetler. Diğer bir deyişle ay 1-12 arası olmalıdır. Eğer girdi geçerli bir ay değilse -1, geçerli bir aysa 0 gönderir.

```
int ay_gecerli_mi(int ay)
{
    if (ay>12 || ay<1) return -1;
    return 0;
}
```

int aydaki_gun(int,int);

Girdi: ay, yıl
Çıktı: [1,31]

(yıl) yılına ait (ay) ayında kaç gün olduğunu bularak geri gönderir. Çıktı 1 ile 31 arasında bir gündür. Bu fonksiyon yıl_artik_mi(int) fonksiyonun da yardımıyla artık yılları da hesaba katar. Fonksiyonda hatalı veri kontrolü yoktur. Girdiler daha önce denetlenmiş olmalıdır.

```
int aydaki_gun(int ay, int yıl)
{
    switch(ay)
    {
        case 2:
            if (yıl_artik_mi(yıl) == 0)
                return 29;
            else
                return 28;
            break;
        case 4:
        case 6:
        case 9:
        case 11:
            return 30;
    }
    return 31;
}
```

void beklet();

Girdi: yok
Çıktı: yok

Tek amacı kullanıldığında kullanıcı bir tuşa basana kadar yazılımı bekletmektir.

```
void beklet()
{
    printf("\nDevam etmek için bir tusa basın.");
    getch();
}
```

int fark_hesapla(int,int,int,int,int,int);

Girdi: ilk gün, ilk ay, ilk yıl, son gün, son ay, son yıl

Çıktı: (0,∞)

İlk ile anlatılmak istenen eski olan tarihtir. Son ile anlatılmak istenen de daha ileri olan tarihtir. Fonksiyon günümüze yakın tarihten eski tarihi çıkararak arada kaç gün geçtiğini bulur ve çıktı olarak gönderir. Fonksiyonda hatalı veri kontrolü yoktur. İlk ile başlayan kısımlara atanan tarih son tarihten büyük olmamalıdır. Girdiler daha önce denetlenmiş olmalıdır.

```
int fark_hesapla(int ilkgun, int ilkay, int ilkyil, int songun, int sonay, int sonyil)
{
    int i;
    int ilk_kacinci_gun=0;
    int son_kacinci_gun=0;

    for (i=1;i<ilkay;i++) // İlk girdide kac gun var?
    {
        ilk_kacinci_gun = ilk_kacinci_gun + aydaki_gun(i,ilkyil);
    }
    ilk_kacinci_gun = ilk_kacinci_gun + ilkgun;

    for(i=1;i<sonay;i++) // Son girdide kac gun var?
    {
        son_kacinci_gun = son_kacinci_gun + aydaki_gun(i,sonyil);
    }
    son_kacinci_gun = son_kacinci_gun + songun;

    if (ilkyil==sonyil) // 00h kebab, aynı yıldan iki güzel gün.
    {
        return son_kacinci_gun - ilk_kacinci_gun;
    }
    else { // Bir de yıllararası gun farki var :( bööööö
        ilk_kacinci_gun = 365 + yıl_artık_mi(ilkyil) + 1 - ilk_kacinci_gun;
        for (i=ilkyil+1;i<sonyil;i++) {
            ilk_kacinci_gun = ilk_kacinci_gun + 365 + yıl_artık_mi(i) + 1;
        }
        return son_kacinci_gun + ilk_kacinci_gun;
    }
}
```

int gun_gecerli_mi(int,int,int);

Girdi: gün, ay, yıl

Çıktı: -1,0

Girdi olan (yıl) yılındaki (ay) ayındaki (gun) gününün geçerli olup olmadığını denetler. Bunu yaparken artık yıl olayını ve ayların kaç çektiklerini de göz önüne alır. Eğer gün geçersizse -1, geçerliyse 0 döndürür.

```
int gun_gecerli_mi(int gun, int ay, int yıl)
{
    int gunsayisi=31;

    if (gun<1 || gun>aydaki_gun(ay,yıl)) return -1;

    return 0;
}
```


void gun_onesi(int,int,int,int);

Girdi: gün, ay, yıl, çıkartılacak gün sayısı

Çıktı: yok

Girdi olan (yıl) yılındaki (ay) ayındaki (gun) gününün (çıkartılacak gün sayısı) kadar gün öncesinin hangi gün olduğunu bulur ve ekrana yazdırır. Geri dönüş değeri yoktur. Fonksiyonda hatalı veri kontrolü yoktur. Fonksiyon 1 Ocak 0001 yılından daha öncesini desteklemez. Programımızda milattan önce desteği bulunmamaktadır.

```
void gun_onesi(int gun, int ay, int yıl, int cikarilacak)
```

```
{
    gun = gun - cikarilacak;
    while (gun<1)
    {
        --ay;
        if (ay<1)
        {
            ay=12;
            --yıl;
            if (yıl<0) { /* Milattan önce desteği vermiyorum */
                printf("Uzgunuz ama su an için milattan oncesi için destegimiz
                bulunmamaktadır.\n");
                return;
            }
        }
        gun = gun + aydaki_gun(ay,yıl);
    }
    printf("\nTarih %d.%d.%d olacaktır.\n",gun,ay,yıl);
}
```

void gun_sonrasi(int,int,int,int);

Girdi: gün, ay, yıl, eklenecek gün sayısı

Çıktı: yok

Girdi olan (yıl) yılındaki (ay) ayındaki (gun) gününün (çıkartılacak gün sayısı) kadar gün sonrasının hangi gün olduğunu bulur ve ekrana yazdırır. Geri dönüş değeri yoktur. Fonksiyonda hatalı veri kontrolü yoktur. gun_onesi(int,int,int,int) fonksiyonuyla görünüş olarak benzerlik gösterse de bu iki fonksiyon içerik itibarıyla tek çatı altında toplanamamıştır.

```
void gun_sonrasi(int gun, int ay, int yıl, int eklenecek)
```

```
{
    gun = gun + eklenecek;
    while (gun>aydaki_gun(ay,yıl))
    {
        gun = gun - aydaki_gun(ay,yıl);
        ++ay;
        if (ay>12)
        {
            ay=1;
            ++yıl;
        }
    }
    printf("\nTarih %d.%d.%d olacaktır.\n",gun,ay,yıl);
}
```

int hangisi_buyuk(int,int,int,int,int,int);

Girdi: birinci gün, birinci ay, birinci yıl, ikinci gün, ikinci ay, ikinci yıl
Çıktı: -2,-1,0

Birinci tarih ile ikinci tarihi karşılaştırır. Eğer birinci tarih büyükse -1, ikinci tarih büyükse 0, iki tarih aynıysa -2 gönderir. Bu fonksiyon fark_hesapla(int,int,int,int,int,int) fonksiyonunda büyük tarihin sona yazılması zorunluluğunu gidermek için yazılmıştır. Bu fonksiyon kullanıcı tarihlerin sırasını ters bile girse (önce büyük bir tarih, sonra günümüze daha uzak bir tarih) bu tarihleri tespit edebileceğimizden değiştirmemizi mümkün kılar. Böylece kullanıcıya hata mesajı vermek yerine ekrana bir uyarı yazdırarak hesaplamamıza devam edebiliriz. Fonksiyonda hatalı veri kontrolü yoktur. Tarih değerleri geçerli olmalıdır.

```
int hangisi_buyuk(int ilkgun, int ilkay, int ilkyil, int songun, int sonay, int sonyil)
{
    // Bu fonksiyonda etkinlik için else yapısına gerek yok. Çünkü if=şartı sağlıyorsa then
    // fonksiyon biter.
    if (sonyil>ilkyil) return 0; // Sorun yok, devam edelim :)
    if (ilkyil>sonyil) return -1; // Eleman normalde geçmiş olması gereken tarihe gelecek
    // tarihi yazmış.
    // E fonksiyon hala işliyorsa o zaman demek ki yıllar aynı.
    // O zaman ne yapıyoruz? Aylara bakıyoruzuuuuuuuuuu!!
    if (sonay>ilkay) return 0;
    if (ilkay>sonay) return -1; //Gıcıklık.
    // Hala mı devam... O zaman artık günlere bakalım.
    if (songun>ilkgun) return 0;
    if (ilkgun>songun) return -1;
    // Ama aynı gün de yazılmaz ki! Ne yapıyoruz, o zaman da -2 gönderiyoruz ki bir çözüm
    // bulsunlar.
    return -2;
}
```

int main()

Girdi: yok
Çıktı: 0

İlk çalışan fonksiyondur. Günün tarihinin istenmesi, menü çizimi, gerekli fonksiyonların çağırılması, veri alımı, menüye geri dönmek, çıkış isteğinin onaylatılması gibi temel olaylar burada olmaktadır. Bu fonksiyonun kaynak kodu oldukça uzun olduğundan birkaç parçada verilmiştir.

```
int main()
{
    /* Bugünün değişkenleri */
    int yıl=-1;
    int ay;
    int gun;

    /* Menü değişkenleri */
    int sag_serbest=-1;
    char secenek=0;

    /* Diğer alınan tarih verileri değişkenleri */
    int ilkyil=-1, ilkay=-1, ilkgun=-1;
    int sonyil=-1, sonay=-1, songun=-1;
    int once_sonra=-1;

    printf("Umut BENZER\n");
    printf("05-06-7670\n");
    printf("Ege Üniversitesi Bilgisayar Muhendisligi 1. Sinif\n");
    printf("http://www.ubenzer.com\n");
    printf("Tarih Hesaplama Yazılımı 1.2\n\n");

    printf("Baslamadan önce sizden bugunun tarihini isteyeceğim.\n");

    while (yil_gecerli_mi(yıl) == -1) /* Niye bu tarih_gecerli mi degil de boyle uc tane
    fonksiyon?? */
    {
        printf("Yıl: ");
        scanf("%d",&yıl); /* Cunku odevde denildi ki, tarih gecersizse
        devamini isteme. */
    }

    while (ay_gecerli_mi(ay) == -1)
    {
        printf("Ay: ");
        scanf("%d",&ay);
    }
}
```

```

while (gun_gecerli_mi(gun,ay,yil) == -1)
{
    printf("Gun: ");
    scanf("%d",&gun);
}

while (sag_serbest==1)
{
    system("cls");
    printf("*** Menu *** (Bugun: %d.%d.%d)\n\n",gun,ay,yil);
    printf("1. Verilen iki tarih arasinda kac gun fark oldugunun bulunmasi \n");
    printf("2. Verilen bir tarihten bugune kadar gecen gun sayisinin bulunmasi \n");
    printf("3. Verilen bir yil icerisinde verilen iki tarih arasinda gecen gun sayisinin bulunmasi \n");
    printf("4. Verilen bir tarihten verilen gun sayisi kadar sonraki tarihin bulunmasi \n");
    printf("5. Bugunden verilen gun sayisi kadar oncesi tarihin bulunmasi \n");
    printf("6. Bugunden verilen gun sayisi kadar sonraki tarihin bulunmasi \n");
    printf("7. Cikis \n\n");

    printf("Lutfen istediginiz islemin numarasini giriniz. ");

    secenek=getche();
    system("cls");

    switch(secenek) /* char fonksiyonu girilen karakterin ASCII kodunu verir.
    { /* Onun icin secenek degiskenini switche sayi olarak sokabiliyo-
    case 49: /* ve case'lere de ASCII kodlarini yerlestiriyorum. :)

        ilkyil=-1;
        ilkay=-1;
        ilkgun=-1; /* Önce sıfırlayalım */
        sonyil=-1;
        sonay=-1;
        songun=-1;
        printf("1. Verilen iki tarih arasinda kac gun fark oldugunun bulunmasi \n");

        while (yil_gecerli_mi(ilkyil) == -1)
        {
            printf("Ilk tarihin yilini giriniz: ");
            scanf("%d",&ilkyil);
        }

        while (ay_gecerli_mi(ilkay) == -1)
        {
            printf("Ilk tarihin ayini giriniz: ");
            scanf("%d",&ilkay);
        }

        while (gun_gecerli_mi(ilkgun,ilkay,ilkyil) == -1)
        {
            printf("Ilk tarihin gununu giriniz: ");
            scanf("%d",&ilkgun);
        }

        while (yil_gecerli_mi(sonyil) == -1)
        {
            printf("Son tarihin yilini giriniz: ");
            scanf("%d",&sonyil);
        }

        while (ay_gecerli_mi(sonay) == -1)
        {
            printf("Son tarihin ayini giriniz: ");
            scanf("%d",&sonay);
        }

        while (gun_gecerli_mi(songun,sonay,sonyil) == -1)
        {
            printf("Son tarihin gununu giriniz: ");
            scanf("%d",&songun);
        }

        switch (hangisi_buyuk(ilkgun,ilkay,ilkyil,songun,sonay,sonyil))
        {
            case -1: // kullanıcı tarih sırasını ters girmiş. Ama biz hata mesajı
            vermiyor, bu hatayı düzelterek kullanıcıya yardımcı oluyoruz. (Yardimsever program)
            printf("\nUYARI: Son tarih, ilk tarihten daha once.\n");
            printf("\nArada %d gun var.",fark_hesapla(songun, sonay, sonyil,
            ilkgun, ilkay, ilkyil));
            break;
            case 0: // sorun yok
            printf("\nArada %d gun var.",fark_hesapla(ilkgun, ilkay, ilkyil, son-
            gun, sonay, sonyil));

```

```

        break;
    case -2: // Haydaaaaaaa
        printf("\nGirdiginiz tarih degerleri ayni. Demek ki gun gecmemis.
:)\n");
    }

    beklet();

    break;
case 50:
    ilkyil=-1;
    ilkay=-1; /* Önce sıfırlayalım */
    ilkgun=-1;

    printf("2. Verilen bir tarihten bugune kadar gecen gun sayisinin bulunmasi
\n");

    while (yil_gecerli_mi(ilkyil) == -1)
    {
        printf("Yil: ");
        scanf("%d",&ilkyil);
    }

    while (ay_gecerli_mi(ilkay) == -1)
    {
        printf("Ay: ");
        scanf("%d",&ilkay);
    }

    while (gun_gecerli_mi(ilkgun,ilkay,ilkyil) == -1)
    {
        printf("Gun: ");
        scanf("%d",&ilkgun);
    }

    switch (hangisi_buyuk(ilkgun,ilkay,ilkyil,gun,ay,yil))
    {
        case -1: // Kullanıcı günümüzden daha ileriki bir tarihi girmiş. Olsun
        varsın, onu da söyleriz biz. (Yardımsever program)
            printf("\n Girdiginiz tarih %d gun SONRASININ tari-
hi.\n",fark_hesapla(gun, ay, yil, ilkgun, ilkay, ilkyil));
            break;
        case 0: // Sorun yok
            printf("\nGirdiniz tarihten bu zamana kadar %d gun
gecmis.",fark_hesapla(ilkgun, ilkay, ilkyil, gun, ay, yil));
            break;
        case -2: // Haydaaaaaaa
            printf("\nBu gunun tarihini girdiginizin farkinda misiniz? Uyklusu-
nuz? :))\n");
    }

    beklet();

    break;
case 51:
    ilkyil=-1;
    ilkay=-1;
    ilkgun=-1; /* Önce sıfırlayalım */
    sonay=-1;
    songun=-1;

    printf("3. Verilen bir yıl içerisinde verilen iki tarih arasında gecen gun
sayisinin bulunmasi \n");

    while (yil_gecerli_mi(ilkyil) == -1)
    {
        printf("Olay hangi yıl icinde gelisecek? ");
        scanf("%d",&ilkyil);
    }

    while (ay_gecerli_mi(ilkay) == -1)
    {
        printf("Ilk tarihin ayini giriniz: ");
        scanf("%d",&ilkay);
    }

    while (gun_gecerli_mi(ilkgun,ilkay,ilkyil) == -1)
    {
        printf("Ilk tarihin gununu giriniz: ");
        scanf("%d",&ilkgun);
    }

    while (ay_gecerli_mi(sonay) == -1)
    {
        printf("Son tarihin ayini giriniz: ");
        scanf("%d",&sonay);
    }

```



```

    }

    while (gun_gecerli_mi(songun,sonay,ilkyil) == -1)
    {
        printf("Son tarihin gununu giriniz: ");
        scanf("%d",&songun);
    }

    switch (hangisi_buyuk(ilkgun,ilkay,ilkyil,songun,sonay,ilkyil))
    {
        case -1: // Kullanıcı tarih sırasını ters girmiş. Ama biz hata mesajı
        vermiyor, bu hatayı düzelterek kullanıcıya yardımcı oluyoruz. (Yardımsever program)
        printf("\nUYARI: Son tarih, ilk tarihten daha önce.\n");
        printf("\nArada %d gun var.",fark_hesapla(songun, sonay, ilkyil,
        ilkgun, ilkay, ilkyil));
        break;
        case 0: // Sorun yok
        printf("\nArada %d gun var.",fark_hesapla(ilkgun, ilkay, ilkyil, son-
        gun, sonay, ilkyil));
        break;
        case -2: // Haydaaaaaaa
        printf ("\ngirdiğiniz tarih degerleri aynı. Demek ki gun gecmemis.
        :)\n");
    }

    beklet();

    break;
case 52:
    ilkyil=-1;
    ilkay=-1; /* Önce sıfırlayalım */
    ilkgun=-1;
    once_sonra=-1;
    printf("4. Verilen bir tarihten verilen gun sayisi kadar sonraki tarihin
    bulunmasi \n");

    while (yil_gecerli_mi(ilkyil) == -1)
    {
        printf("Yil: ");
        scanf("%d",&ilkyil);
    }

    while (ay_gecerli_mi(ilkay) == -1)
    {
        printf("Ay: ");
        scanf("%d",&ilkay);
    }

    while (gun_gecerli_mi(ilkgun,ilkay,ilkyil) == -1)
    {
        printf("Gun: ");
        scanf("%d",&ilkgun);
    }

    while (gun_gecerli_mi(ilkgun,ilkay,ilkyil) == -1)
    {
        printf("Gun: ");
        scanf("%d",&ilkgun);
    }

    while (once_sonra < 0)
    {
        printf("Kac gun sonrasini hesaplamami istersiniz? ");
        scanf("%d",&once_sonra);
    }

    gun_sonrasi(ilkgun,ilkay,ilkyil,once_sonra);
    beklet();

    break;
case 53:
    printf("5. Bugunden verilen gun sayisi kadar onceki tarihin bulunmasi \n");
    once_sonra=-1; /* Önce sıfırlayalım */

    while (once_sonra < 0)
    {
        printf("Bugunden kac gun oncesini hesaplamami isterdiniz? ");
        scanf("%d",&once_sonra);
    }

    gun_onesi(gun,ay,yil,once_sonra);
    beklet();

```

```

        break;
    case 54:
        printf("6. Bugunden verilen gun sayisi kadar sonraki tarihin bulunmasi \n");
        once_sonra=-1; /* Önce sıfırlayalım */
        while (once_sonra < 0)
        {
            printf("Bugunden kac gun sonrasini hesaplamami isterdiniz? ");
            scanf("%d",&once_sonra);
        }

        gun_sonrasi(gun,ay,yil,once_sonra);

        beklet();

        break;
    case 65:
        printf("Supriz Yumurta | Easter Egg\n");
        printf("I would love to change the world, but they won't give me the source
code.\n");
        printf("http://en.wikiquote.org/wiki/Computers#Programming (Ano-
nim)\n\n\n");
        printf("Tebrikler, bu programin gizli ozelligini buldunuz. :)\n Devam etmek
icin bir tusa basin.\n");
        getch();
        break;
    case 55:
        do {
            printf ("\nCikmak istediginizden emin misiniz? (E-H)");
            secenek = getch();
            if (secenek == 'E' || secenek == 'e') {
                sag_serbest = 0;
                secenek = 'H';
            }
        } while (secenek != 'H' && secenek != 'h');

        break;
    default:
        printf("Lutfen 1'den 7'ye kadar bir secim yapin.",secenek);
        beklet();
        break;
    }
}

printf("\n\nProgram sonlandi. :)");
return 0;
}

```

int yil_artik_mi(int);

Girdi: yıl
Çıktı: -1,0

Girdi olan (yıl) yılının artık bir yıl olup olmadığını kontrol eder. Artık yıl dörde bölümünden sıfır kalan yıllar olarak tanımlanabilir. Program genelinde -1 HAYIR, 0 EVET olarak alındığından eğer yıl artıksa 0, değilse -1 değeri döndürülür. Fonksiyonda hatalı veri kontrolü yoktur. Fonksiyona gönderilecek değer önceden kontrolü yapılmış değerler olmalıdır.

int yil_artik_mi(int yıl)

```

{
    /* Bir yılın 4'e bölümünden kalan 0 ise o yıl artıktır. */
    /* Ancak o yıl 100'e bölünüyorsa artık değildir. Ama 400'e de bölünüyorsa artıktır. */
    if ((yıl % 4 == 0 && yıl % 100 != 0) || (yıl % 4 == 0 && yıl % 400 == 0)) return 0;
    return -1;
}

```

int yil_gecerli_mi(int)

Girdi: yıl
Çıktı: -1,0

Girdi olan (yıl) yılının geçerli olup olmadığını denetler. Eğer yıl geçerliyse 0, geçersizse 1 değeri döndürülür. Milattan öncesi (yani – değerler) geçersiz sayılmaktadır.

int yil_gecerli_mi(int yıl)

```

{
    if(yıl<1) return -1; /* Milattan önce desteğimiz bulunmamaktadır. */
    return 0;
}

```

Kullanıcı Kataloğu

Bir Ubenzer yazılımı olan Tarih Hesaplama Yazılımını tercih ettiğiniz için teşekkürler. Tarih Hesaplama Yazılımı kullanma kılavuzu aradığınız bölüme daha kolay erişebilmeniz için başlıklar halinde sınıflandırılmıştır.

1. Kurulum
2. Kullanım
3. Daha fazlası...
4. Kısıtlamalar
5. Çevrimiçi yardım

Kurulum

Programın derlenmiş hali Tarih Hesaplama.exe isminde dağıtılmaktadır. Programı çalıştırmak için herhangi bir kurulum söz konusu değildir. Size ulaştırılan Tarih Hesaplama.exe uygulamasına çift tıklayarak programı başlatabilirsiniz.

Programı kaynak koduyla çalıştırmak isteyen kullanıcılarımız ise kendilerine verilen C# kodunu uygun bir C derleyicisinde derleyerek programı açabilirler. Popüler C# derleyicilerinin bir listesini Wikipedia'nın ilgili makalesinde bulabilirsiniz:

http://en.wikipedia.org/wiki/List_of_compilers#C.2FC.2B.2B_compilers (Bu makale İngilizce olabilir.)

Kullanım

Programı başlattığınızda sizden ilk olarak içinde bulunduğumuz günün tarihi istenecektir. Günümüzün tarihi programda istendiği şekilde Yıl, Ay, Gün sırasına göre girilmelidir. Geçersiz bir tarih girdiğinizde (örneğin yıl için -2 yazıldığında veya ay için 13 yazıldığında) program bir sonraki veriyi istemek yerine doğru yazılana kadar aynı veriyi isteyecektir. Programa tüm yapılacak tüm girdilerin sayı olması büyük önem taşımaktadır.

Program günümüzün tarihini aldıktan sonra ana menüyü görüntüler. Menü yazısının sağ tarafında günümüzün tarihinin görünmesi programın kullanılmasında kolaylık sağlamaktadır. Bu noktada devam etmek için tek yapmanız gereken menüde görebileceğiniz seçenekler içerisinde hangisini istiyorsanız o seçeneğin sağında görebileceğiniz rakama klavye ile basmaktır. Bu esnada aslında menüde olmayan bir seçeneğe basarsanız program bunu algılayarak seçenekleri size tekrar sunar. Böylece menü hep ekranda kalırken, kullanıcıya da büyük kolaylık sağlanmış olur.

Bu yazının ilerleyen kısımlarında menüdeki her seçeneğin nasıl kullanıldığı ayrıntılı bir şekilde anlatılacaktır.

1. VERİLEN İKİ TARİH ARASINDA KAÇ GÜN OLDUĞUNUN BULUNMASI

Bu özellik kullanıcıdan istenecek iki tarih arasında kaç gün olduğunu bulur. Bu özelliğe menüde 1'i tuşlayarak erişebilirsiniz. Özelliğe girdikten sonra program sizden birinci tarihin sırasıyla yıl, ay ve gün değerlerini isteyecektir. Eğer girdiğiniz verilerde geçersizlik durumu saptanırsa program bir sonraki veriye geçmeden önce hatalı veriyi tekrar ister. Birinci tarih girildikten sonra program sizden ikinci tarihi de yine aynı sırayla isteyecektir. Program tarihleri aralarında karşılaştırabilme yeteneğine sahip olduğu için önce geçmiş, sonra günümüze daha yakın tarihi girme gibi bir zorunluluğunuz bulunmamaktadır. Program her iki şekilde girilen tarihi de sorunsuz hesaplayabilir.

Hesaplama bittikten sonra program menüye döner. Özelliği tekrar kullanmak için tekrar 1'e basmanız yeterli olacaktır.

2. VERİLEN BİR TARİHTEN BUGÜNE KADAR GEÇEN GÜN SAYISININ BULUNMASI

Bu özellik kullanıcıdan istenecek bir tarih ile günümüz arasında kaç gün olduğunu bulur. Özelliğe menüde 2'yi tuşlayarak erişebilirsiniz. Özelliğe girdikten sonra program sizden bugün ile arasındaki gün sayısı bulunacak tarihi sırasıyla yıl, ay ve gün olarak isteyecektir. Eğer girdiğiniz verilerde geçersizlik durumu saptanırsa program bir sonraki veriye geçmeden önce hatalı veriyi tekrar ister. Eğer gelecekte bir tarih yazmışsanız program ekranda bunu bildirir ve yazdığınız tarihin kaç gün sonrasına ait olduğunu söyler.

Hesaplama bittikten sonra program menüye döner. Özelliği tekrar kullanmak için tekrar 2'ye basmanız yeterli olacaktır.

3. VERİLEN BİR YIL İÇERİSİNDE VERİLEN İKİ TARİH ARASINDA GEÇEN GÜN SAYISININ BULUNMASI

Bu özellik ile aynı yıl içerisindeki iki tarih arasında kaç gün olduğunu kolayca bulabilirsiniz. Özelliğe menüde 3'ü tuşlayarak erişebilirsiniz. Özelliğe girdikten sonra program sizden öncelikle gireceğiniz iki tarihin hangi yıla ait olduğunu soracaktır. Bunu artık yılı da hesaba katmak için yapıyoruz. Daha sonra sizden sırasıyla birinci ve ikinci tarihi (ay, gün sırasına göre) isteyecektir. Eğer girdiğiniz verilerde geçersizlik durumu saptanırsa program bir sonraki veriye geçmeden önce hatalı veriyi tekrar ister. Program tarihleri karşılaştırma yeteneğine sahip olduğundan tarihleri kronolojik sıraya göre yazma zorunluluğunuz bulunmamaktadır.

Hesaplama bittikten sonra program menüye döner. Özelliği tekrar kullanmak için tekrar 3'e basmanız yeterli olacaktır.

4. VERİLEN BİR TARİHTEN VERİLEN GÜN KADAR SONRASININ TARİHİNİN BULUNMASI

Bu özellik kullanıcıdan istenecek bir tarihten yine kullanıcıdan istenecek gün kadar sonrasının hangi tarihe denk geldiğini bulmaktadır. Özelliğe menüde 4'ü tuşlayarak erişebilirsiniz. Özelliğe girdikten sonra program sizden yıl, ay, gün sırasında olmak üzere istediğiniz gün sonrasını bulmak istediğiniz tarihi isteyecektir. Verileri aldıktan sonra kaç gün sonrasını bulmak istediğinizi soracaktır. Lütfen unutulmamalıdır ki bu girdi eksi değer alamaz. Eğer girdiğiniz verilerde geçersizlik durumu saptanırsa program bir sonraki veriye geçmeden önce hatalı veriyi tekrar ister. Tüm girdiler tamamlandıktan sonra girdiğiniz tarihten girdiğiniz gün kadar sonrasının tarihi ekrana yazdırılır.

Hesaplama bittikten sonra program menüye döner. Özelliği tekrar kullanmak için tekrar 4'e basmanız yeterli olacaktır.

5. BUGÜNDEN VERİLEN GÜN KADAR ÖNCESİNİN TARİHİNİN BULUNMASI

Bu özellik bugünden kullanıcıdan istenecek gün sayısı kadar öncesinin hangi tarihe geldiğini bulur. Özelliğe menüde 5'i tuşlayarak erişebilirsiniz. Özelliğe girdikten sonra program kaç gün öncesinin bulunmasını istediğinizi soracaktır. Eğer girdiğiniz veride geçersizlik durumu saptanırsa program veriyi tekrar ister. Bilgi girişi tamamlandığında program tarihi ekrana yazdıracaktır. Bu özellikteki kısıtlamalar için kısıtlamalar kısmını okuyabilirsiniz.

Hesaplama bittikten sonra program menüye döner. Özelliği tekrar kullanmak için tekrar 5'e basmanız yeterli olacaktır.

6. BUGÜNDEN VERİLEN GÜN KADAR SONRASININ TARİHİNİN BULUNMASI

Bu özellik bugünden kullanıcıdan istenecek gün sayısı kadar sonrasının hangi tarihe geleceğini bulur. Özelliğe menüde 6'yı tuşlayarak erişebilirsiniz. Özelliğe girdikten sonra program kaç gün sonrasının bulunmasını istediğinizi soracaktır. Eğer girdiğiniz veride geçersizlik durumu saptanırsa program veriyi tekrar ister. Bilgi girişi tamamlandığında program tarihi ekrana yazdıracaktır.

Hesaplama bittikten sonra program menüye döner. Özelliği tekrar kullanmak için tekrar 6'ya basmanız yeterli olacaktır.

7. ÇIKIŞ

Programı kapatmanızı sağlar. Özelliğe menüde 7'yi tuşlayarak erişebilirsiniz. Program kapanmadan önce sizden onay bekleyecektir. Kapatmak istediğinize eminseniz ekrandaki soruyu "E" ile cevaplayarak kapatabilir ya da başka bir tuşa basarak menüye geri dönebilirsiniz.

Kısıtlamalar

Program kullanılırken dikkat edilmesi gereken bazı önemli noktalar mevcuttur. Tarih Hesaplama yazılımı bu sürümünde milattan öncesi için destek vermemektedir. Kullanıcıdan veri alırken negatif yıl değerlerini kabul etmemekte ve bugünden verilen gün sayısı kadar öncesini bulma işleminde hesaplama sonucu 1 Ocak 0001 gününden daha öncelere düşüyorsa gerekli uyarı mesajı ile hesaplama reddedilmektedir. Programa yapılan tüm girdiler sayısal olmalıdır. Program sayısal girdilerdeki hataları bulacak ve doğrusunu isteyecek şekilde tasarlanmıştır. Ancak programımızda değişken tipindeki uyumsuzluktan oluşan hatalara karşı bir ayıklama mevcut değildir. Söz gelimi, Yıl: asfj gibi bir girdi programın çalışmamasına neden olacaktır. Böyle bir şey yapıldığında programı tekrar çalışabilir hale getirmek için programı kapatıp tekrar başlatmalısınız.

Daha Fazlası



Programda kullanım kılavuzunda anlatılan dışında gizli bir özellik daha mevcuttur. Bu özellik programın (ana menü de dâhil olmak üzere) hiçbir yerinde gösterilmemiştir. Özelliğe ana menüdeyken klavyeden bir tuşa basarak erişilebilir. Bu özelliği keşfetmeyi bu kullanım kılavuzunu okuyan siz değerli müşterilerimize bırakıyoruz.

Çevrimiçi Yardım

Eğer daha fazlasına ihtiyacınız varsa çevrimiçi yardım alabilirsiniz. Bunun için <http://www.ubenzer.com/iletisim> adresindeki iletişim formunu kullanarak bize ulaşın.