1. Başla.
2. Struct tipinde bir kayıt değişkeni oluştur.

Int tipinde OgrNo,dersKodu,puan; yirmişer elemanlı char tipinde ad,soyad değişkenleri oluştur.

1. Main fonksiyonu oluşturulur.
   1. Char pointer tipinde dataFilePath ve indexFilePath değişkenleri oluşturuldu ve pathleri girildi.
   2. Int tipinde “ k ” değişkeni oluşturuldu, değeri 1 olarak atandı.
   3. Char tipinde continueChar değişkeni oluşturuldu.
   4. k eşit değil 0 olduğu sürece çalışan while döngüsü oluşturuldu
      * “Öğrenci kayıt programına hoş geldiniz” yazdırır.
      * “Lütfen bir seçim yapınız.” yazdırır.
      * “İndex dosyası oluşturmak için: 1 giriniz.” yazdırır.
      * “Kayıt eklemek için: 2 giriniz” yazdırır.
      * “Kayıt bulmak için: 3 giriniz.” yazdırır.
      * “Kayıt silmek için: 4 giriniz.” yazdırır.
      * “Kayıt güncellemek için: 5 giriniz.” yazdırır.
      * “Veri dosyasını görüntülemek için: 6 giriniz.” yazdırır.
      * “İndex dosyasını görüntülemek için: 7 giriniz.” yazdırır.
      * “İndex dosyasını silmek için: 8 giriniz.” yazdırır.
      * “Programı kapatmak için: 0 giriniz.” yazdırır.
      * “Giris: “yazdırır.
      * Kullanıcıdan k değişkeni için bir değer alır.
      * K değişkenine bağlı bir switch – case yapısı oluşturulur.
        1. Case 0 için
        2. “Program sonlandı. İyi günler” yazdırır.
        3. Switch-case yapısından çıkar.
        4. Case 1 için
           1. indexOlustur fonksiyonunu DataFilePath’i göndererek çalıştırır.
           2. Switch-case yapısından çıkar.
        5. Case 2 için
           1. kayitEkle fonksiyonuna dataFilePath ve indexFilePath değişkenlerini göndererek çalıştırır.
           2. Switch-case yapısından çıkar.
        6. Case 3 için
           1. kayitBul fonksiyonuna dataFilePath ve indexFilePath değişkenlerini göndererek çalıştırır.
           2. Switch-case yapısından çıkar.
        7. Case 4 için
           1. kayitSil fonksiyonuna dataFilePath ve indexFilePath değişkenlerini göndererek çalıştırır.
           2. Switch-case yapısından çıkar.
        8. Case 5 için
           1. kayitGuncelle fonksiyonuna dataFilePath ve indexFilePath değişkenlerini göndererek çalıştırır.
           2. Switch-case yapısından çıkar.
        9. Case 6 için
           1. veriDosyasiniGoster fonksiyonuna dataFilePath değişkenini göndererek çalıştırır.
           2. Switch-case yapısından çıkar.
        10. Case 7 için
            1. indexDosyasiniGoster fonksiyonuna indexFilePath değişkenini göndererek çalıştırır.
            2. Switch-case yapısından çıkar.
        11. Case 8 için
            1. indexDosyasiniSil fonksiyonuna indexFilePath değişkenini göndererek çalıştırır.
            2. Switch-case yapısından çıkar.

* + - 1. Default
         1. “Gecersiz Giris” yazdırır.
         2. Switch-case yapısından çıkar.

1. indexOlustur fonksiyonu oluşturulur.
   1. FILE \* tipinde dataFile değişkeni oluşturulur.
   2. FILE \* tipinde indexFile değişkeni oluşturulur.
   3. Int tipinde count değişkeni oluşturulur.
      * 0 değeri atanır.
   4. Int \* tipinde datakeys değişkeni oluşturulur.
   5. Int \* tipinde offsets değişkeni oluşturulur.
   6. Long int tipinde filesize değişkeni oluşturulur.
      * 0 değeri atanır.
   7. Int tipinde recordCount değişkeni oluşturulur.
      * 0 değeri atanır.
   8. Struct kayit \* tipinde ogr değişkeni oluşturulur.
   9. Int tipinde offset değişkeni oluşturulur.
   10. size\_t tipinde readElementCount değişkeni oluşturulur.
   11. dataFilePath rb (read binary) modunda açılır.
       * Oluşturulan dosya dataFile değişkenine atanır.
   12. dataFile değişkenine atanmış dosyanın sonuna konumlanılır.
   13. fileSize değişkenine dataFile değişkeninin boyutu atanır.
   14. recordCount değişkenine (fileSize / struct kayit değişkenin boyutu) sonucu atanır.
   15. datakeys değişkenine bellekten fileSize kadar yer ayrılır.
   16. offsets değişkenine bellekten recordCount \* 4 byte yer ayrılır.
   17. datakeys değişkeninin bellek adresi boşaltılır.
   18. indexFile değişkenine, verilen pathte w+ modunda açılan dosya atanır.
   19. dataFile’ın başına konumlanılır.
   20. KayitOku fonksiyonu ogr değişkenine aktarılabildiği sürece çalışan bir while döngüsü oluşturulur.
       * offset değişkenine (count\* (kayıt structının boyutu)) değeri atanır.
       * datakeysin count indexine ogr structının ogrNo değişkeni atanır.
       * offetsin count indexine offsetin değeri atanır.
       * count 1 artırılır.
       * t’ye dönülür.
   21. SortKayit fonksiyonuna datakeys, offets, recordCount değişkenleri gönderilir ve çalıştırılır.
   22. count=0’dan başlayıp count recordCount’tan küçük olduğu sürece count’ın 1 artırıldığı for döngüsü oluşturulur. (for (count=0;count<recordCount;count++))
       * datakeys ve offsetsin count indexindeki değişkenler sırasıyla indexFile’a yazdırılır.
   23. dataFile dosyası kapatılır.
   24. indexFile dosyası kapatılır.
2. SortKayit fonksiyonu oluşturulur.
   1. SortKayit fonksiyonunun aldığı değerler int \*Kayitlar, int \*offsets, int \*kayitSayisi.
   2. Int tipinde i,j,swapKayit ve swapOffset değişkenleri tanımlanır.
   3. i=0 için i kayitSayisi -1’den küçük olduğu sürece i’nin 1 arttığı for döngüsü oluşturulur.
      * For döngüsünün içine j=i+1 için j kayitSayisi’ndan küçük olduğu sürece j’nin 1 arttığı for döngüsü oluşturulur.
        1. Kayitlar dizisinin i. indisi kayitlar dizisinin j. indisinden büyükse
           1. Kayitlar dizisinin j. indisi swapKayit değişkenine atanır.
           2. Kayitlar dizisinin i. indisi Kayitlar dizisinin j. indisine atanır.
           3. swapKayit değişkeni Kayitlar dizisinin i. indisine atanır.
           4. offsets dizisinin j. indisi swapOffset değişkenine atanır.
           5. offsets dizisinin i. indisi offsets dizisinin j. indisine atanır.
           6. swapOffset değişkeni offsets dizisinin i. değişkenine atanır.
3. KayitOku fonksiyonu oluşturulur.
   1. FILE \* tipinde dataFile isminde değişken alır.
   2. Opak pointer tipinde buffer değişkeni tanımlanır.
   3. Sturct kayit \* tipinde ogr değişkeni tanımlanır.
   4. size\_t tipinde readElementCount değişkeni tanımlanır.
   5. buffer değişkenine bellekten kayıt structının boyutu kadar yer ayrılır.
   6. buffera ayrılan yer temizlenir.
   7. dataFile değişkeninden kayıt structının boyutu kadar byte buffer’a yazılır ve işlemin sonucuna göre readElementCount değişkenine sonuç atanır.
   8. readElementCount = 0 için
      * buffer için tutulan bytelar serbest bırakılır.
      * NULL değer döndürülür.
   9. ogr değişkeni için bellekten kayıt structı kadar yer ayrılır.
   10. bufferın içerisindeki bytelar kayıt structı formatına çevrilerek ogr değişkenine yazdırılır.
   11. buffer serbest bırakılır.
   12. ogr structı geri döndürülür.
4. kayitEkle fonksiyonu oluşturulur.
   1. char \* tipinde dataFilePath ve indexFilePath değişkenleri fonksiyona gönderilir.
   2. FILE \* tipinde indexFile ve dataFile değişkenleri oluşturulur.
   3. indexFilePath şeklinde pathi verilen dosya a+ modunda indexFile’da açılır.
   4. dataFilePath şeklinde pathi verilen dosya ab+ modunda dataFile’da açılır.
   5. struct kayit tipinde o değişkeni oluşturulur.
   6. “Öğrenci no giriniz:” yazdırılır.
   7. o structının ogrNo değişkeninin girişi alınır.
   8. “Ders kodu giriniz:” yazdırılır.
   9. o structının dersKodu değişkeninin girişi alınır.
   10. “Puan giriniz:” yazdırılır.
   11. o structının puan değişkeninin girişi alınır.
   12. o structının ad değişkeninde tutulan bytelar temizlenir.
   13. “Ad giriniz:” yazdırılır.
   14. o structının ad değişkeninin girişi alınır.
   15. o structının soyad değişkeninde tutulan bytelar temizlenir.
   16. “Soyad giriniz:” yazdırılır.
   17. o sturctının soyad değişkeninin girişi alınır.
   18. o structı dataFile’a yazdırılır.
   19. dataFile kapatılır.
   20. indexFile kapatılır.
   21. indexOlustur fonksiyonu çağrılır.
5. indexDosyasiniGoster fonksiyonu oluşturulur.
   1. char \* tipinde indexFilePath değişkeni fonksiyona gönderilir.
   2. int tipinde ogrNo ve offset değişkenleri tanımlanır.
   3. FILE \* tipinde indexDosyasi değişkeni tanımlanır.
   4. indexFilePath r+ modunda indexDosyasi değişkenine açılır.
   5. indexDosyasi = NULL olduğunda:
      * “Dosya bulunamadi.” yazdırılır.
      * main fonksiyona geri dönülür.
   6. indexDosyasinin başına konumlanılır.
   7. “Ogrenci listesi:” yazdırılır.
   8. indexDosyasinin sonuna gelinmediği sürece
      * indexDosyasindan okunan değerler sırasıyla ogrNo ve offset değişkenlerine atanır.
      * “Ogrenci no: offset:” yazdırılır.
      * indexDosyasi kapatılır.
6. veriDosyasiniGoster fonksiyonu oluşturulur.
   1. char \* tipinde dataFilePath değişkeni fonksiyona gönderilir.
   2. FILE \* tipinde dataFile değişkeni oluşturulur.
   3. sturct kayıt \* tipinde ogr değişkeni oluşturulur.
   4. int count değişkeni 1 olarak atanır.
   5. dataFilePath rb modunda dataFile değişkenine açılır.
   6. dataFile’ın başına konumlanılır.
   7. KayitOku fonksiyonu ogr değişkenine atanabildiği sürece
      * “(count). öğrenci”yazdırılır.
      * “Ogrenci no: (ogrNo)” yazdırılır.
      * “Ders kodu: (dersKodu)” yazdırılır.
      * “Puani: (puan)” yazdırılır.
      * “Adi: (ad)” yazdırılır.
      * “Soyadi: (soyad)” yazdırılır.
      * count değişkeni 1 artırılır.
   8. dataFile kapatılır.
7. indexDosyasiniSil fonksiyonu oluşturulur.
   1. char \* tipinde fileName değişkeni fonksiyona gönderilir.
   2. fileName değişkeninde tutulan dosya silinir.
   3. “Index dosyasi silindi.” yazdırılır.
8. kayitBul fonksiyonu oluşturulur.
   1. char \* dataFilePath ve indexFilePath değişkenleri gönderilir.
   2. FILE \* dataFile ve indexFile değişkenleri tanımlanır.
   3. struct kayıt \* tipinde ogr değişkeni tanımlanır.
   4. Opak \* buffer değişkeni tanımlanır.
   5. int tipinde ogrNo, ogrNo1, offset1, line=0, c=0, mid, last, first=0 değişkenleri tanımlanır.
   6. dataFile değişkeninde dataFilePathteki dosya rb modunda açılır.
   7. indexFile değişkeninde indexFilePathteki dosya r modunda açılır.
   8. “Bulmak istediğiniz öğrencinin numarasi:” yazdırılır.
   9. Kullanıcıdan ogrNo değişkeni için giriş alınır.
   10. indexFile’ın başına konumlanılır.
   11. indexFile’ın sonuna gelinmediği sürece devam eden while döngüsü oluşturulur.
       * indexFile’dan ogrNo1 ve offset1 değişkenleri okunabiliyorsa
         1. line değişkeni 1 artırılır.
   12. line değişkeni kadar elemanı olan int No[] ve offset[] dizileri tanımlanır.
   13. indexFile’ın başına konumlanılır.
   14. indexFile’ın sonuna gelinmediği sürece devam eden while döngüsü oluşturulur.
       * indexFile’dan okunan değerler sırasıyla No[] dizisinin c. indisine ve offset[] dizisinin c. indisine atanır.
       * c 1 artırılır.
   15. last değişkenine line değişkeninin 1 eksiği atanır. (last=line-1)
   16. mid değişkenine last ve first değişkenlerinin toplamının yarısı değer olarak atanır.(mid=(last+first)/2)
   17. first değişkeni last değişkenine eşit veya last değişkeninden küçük olduğu sürece devam eden while döngüsü oluşturulur.
       * No[] dizisinin mid. indisi ogrNo değişkeninden küçükse
         1. first değişkenine mid değişkeninin 1 fazlası atanır. (first=mid+1)
       * No[] dizisinin mid. indisi ogrNo değişkenine eşitse
         1. döngüden çıkılır.
       * Bu şartlar sağlanmıyorsa
         1. last değişkenine mid değişkeninin değerinin 1 eksiği atanır.(last=mid-1)
       * mid değişkeni first ve last değişkenlerinin değerleri toplamının yarısı değer olarak atanır. (mid=(first+last)/2)
   18. int tipinde up değişkeni oluşturulur ve 0 değeri atanır.
   19. No[] dizisinin (mid-up). indisi ogrNo’ya eşit olduğu sürece devam eden while döngüsü oluşturulur.
       * up değişkeni 1 artırılır.
   20. up değişkeni 1 azaltılır.
   21. int tipinde tf değişkeni oluşturulur.
   22. Sonsuz dönen while döngüsü oluşturulur.
   23. No[] dizisinin (mid-up). indisi ogrNo değişkenine eşitse
       * buffer’a ve ogr’ye bellekten kayıt structı kadar yer ayrılır.
       * buffer değişkenine ayrılan yerler boşaltılır.
       * dataFile’ın imleci offset [] dizisinin (mid-up). indisindeki offsete konumlanır.
       * dataFile’dan kayıt structının boyutu kadar byte buffer’a yazılır.
       * buffer kayıt structı şeklinde biçimlendirilir ve oluşan yapı ogr değişkenine aktarılır.
       * “Aradiginiz öğrenci kaydi aşağıdaki mi?” yazdırılır.
       * “Ogrenci no: (ogrNo) ” yazdırılır.
       * “Ders kodu: (dersKodu)” yazdırılır.
       * “Puani: (puan)” yazdırılır.
       * “Adi: (ad)” yazdırılır.
       * “Soyadi: (soyad)” yazdırılır.
       * “Buysa 1 degilse 0 giriniz: ” yazdırılır.
       * tf değişkenine kullanıcıdan alınan değer atanır.
   24. Bu şart sağlanmıyorsa
       * “Aradiginiz kayit bulunmamaktadir” yazdırılır.
       * ogr değişkenine ayrılan byte serbest bırakılır.
       * buffer değişkenine ayrılan byte serbest bırakılır.
       * 1 değeri döndürülür.
   25. tf değişkeni 0’a eşitse
       * up değişkenin değeri 1 azaltılır.
       * while döngüsünün başına dönülür.
   26. tf değişkeni 1’e eşitse
       * ogr değişkenine ayrılan byte serbest bırakılır.
       * buffer değişkenine ayrılan byte serbest bırakılır.
       * döngüden çıkılır.
   27. dataFile kapatılır.
   28. indexFile kapatılır.
   29. offset[] dizisinin (mid-up). indisi değer olarak döndürülür.
9. kayitSil fonksiyonu oluşturulur.
   1. char \* tipinde dataFilePath ve indexFilePath değişkenleri fonksiyona gönderilir.
   2. FILE \* tipinde dataFile değişkeni oluşturulur.
   3. FILE \* tipinde indexFile değişkeni oluşturulur.
   4. Opak \* tipinde buffer değişkeni oluşturulur.
   5. buffer değişkenine bellekten kayıt structı kadar yer ayrılır.
   6. dataFile değişkenine dataFilePath değişkeninde pathi verilen dosya r+b modunda açılır.
   7. indexFile değişkenine indexFilePath değişkeninde pathi verilen dosya r+ modunda açılır.
   8. int tipinde offset ve delete = 1 değişkenleri tanımlanır.
   9. offset değişkenine kayitBul fonksiyonundan dönen değer atanır.
   10. offset 1’e eşitse
       * kayitSil fonksiyonu boş değer döndürür.
   11. dataFile’da offset + (52 \* delete) konumuna gelinir.
   12. Sonsuz bir while döngüsü oluşturulur.
       * buffer değişkeninin içerisindeki bytelar temizlenir.
       * dataFile’dan kayıt structı kadar byte buffer’a aktarılabiliyorsa
         1. dataFile’da offset + ((delete-1)\*52) konumuna gelinir.
         2. buffer dataFile’a yazdırılır.
         3. dataFile’da offset + (52\* (delete+1)) konumuna gelinir.
         4. delete 1 artırılır.
       * Yapılamıyorsa
         1. int tipindeki size değişkenine offset + (52\*(delete-1)) değeri atanır.
         2. dataFilePath değişkeninin boyutu size değişkenine indirgenir.
         3. döngüden çıkılır.
   13. buffer serbest bırakılır.
   14. indexOlustur fonksiyonu “./files/data.bin” gönderilerek çağrılır.
   15. dataFile kapatılır.
   16. indexFile kapatılır.
10. kayitGuncelle fonksiyonu oluşturulur.
    1. char \* tipinde dataFilePath ve indexFilePath değişkenleri fonksiyona gönderilir.
    2. FILE \* tipinde dataFile değişkeni oluşturulur.
    3. FILE \* tipinde indexFile değişkeni oluşturulur.
    4. int tipinde not ve offset değişkenleri oluşturulur.
    5. dataFile değişkenine dataFilePath değişkeninde pathi verilen dosya r+b modunda açılır.
    6. indexFile değişkenine indexFilePath değişkeninde pathi verilen dosya r+ modunda açılır.
    7. offset değişkenine kayitBul fonksiyonuna dataFilePath ve indexFilePath gönderildikten sonraki dönen değer atanır.
    8. offset = 1 ise
       * kayitGuncelle fonksiyonu değer döndürmez.
    9. “Ogrencinin yeni puanini giriniz: ” yazdırılır.
    10. Kullanıcıdan not değişkenine değer alınır.
    11. dataFile’da offset + 8 konumuna gelinir.
    12. not değişkeni dataFile’a yazılır.
    13. “Ogrencinin yeni puanı: (not)” yazdırılır.
    14. dataFile kapatılır.
    15. indexFile kapatılır.
11. Bitir.