

Ad Soyad

No :

1	2	3	4
---	---	---	---

1. Kitap veritabanı aşağıdaki üyelerden oluşur.

index (int), bookName (101), authorName (51), pubYear (int) ve isbnNumber (21)

Bu bilgilere göre:

- a. Bu bilgileri içeren bir **“BookDatabase”**
- b. isimli bir struct tanımlayın.(5 Puan)
- c. Main fonksiyonunda, kullanıcıdan alınan n tane kitap için BookDatabase için dinamik bellek ayırın. (5 Puan)
- c. Prototipi **void addBook(int index, BookDatabase *book)** olan, kullanıcıdan bir kitabı bilgilerini (**bookName, authorName, pubYear, isbnNumber**) alan ve bu bilgileri veri tabanına index'inci eleman olarak atan bir fonksiyon yazınız. (10 Puan)
- d. Veri tabanındaki tüm kitap (n sayısı) bilgilerini görüntüleyen **void display(int n, BookDatabase *book)** prototipine sahip bir fonksiyon yazın. (5 Puan)

Uyarı: Stringleri atamak için, "string.h" içinde tanımlanan **strcpy, strlen, vb. kullanmayın.**

Bunları kullanmak yerine, kaynağı hedefe kopyalayan void copy(char *source,char *destination) prototipli bir fonksiyon tanımlayın.

```
#include <stdio.h>
#include <stdlib.h>

#define MAX_BOOK_NAME 101
#define MAX_AUTHOR_NAME 51
#define MAX_ISBN_NUMBER 21

// Kitap veritabanı için struct tanımı
typedef struct {
    int index;
    char bookName[MAX_BOOK_NAME];
    char authorName[MAX_AUTHOR_NAME];
    int pubYear;
    char isbnNumber[MAX_ISBN_NUMBER];
} BookDatabase;

// Kaynaktan hedefe string kopyalama fonksiyonu
void copy(char *source, char *destination) {
    while (*source) {
        *destination = *source;
        source++;
        destination++;
    }
    *destination = '\0'; // hedefin sonunu belirleyelim
}
```

```
// Kullanıcıdan bir kitabın bilgilerini alıp veri tabanına ekleyen fonksiyon
void addBook(int index, BookDatabase *book) {
    book->index = index;

    printf("Kitap adı: ");
    scanf("%100s", book->bookName); // %100s ile buffer overflow riskini azaltıyoruz

    printf("Yazar adı: ");
    scanf("%50s", book->authorName); // %50s ile buffer overflow riskini azaltıyoruz

    printf("Yayın yılı: ");
    scanf("%d", &book->pubYear);

    printf("ISBN numarası: ");
    scanf("%20s", book->isbnNumber); // %20s ile buffer overflow riskini azaltıyoruz
}

// Veritabanındaki tüm kitap bilgilerini görüntüleyen fonksiyon
void display(int n, BookDatabase *book) {
    printf("\nKitap Bilgileri:\n");
    printf("Index\tKitap Adı\tYazar Adı\tYayın Yılı\tISBN Numarası\n");
    for (int i = 0; i < n; i++) {
        printf("%d\t%s\t%s\t%d\t%s\n",
               book[i].index, book[i].bookName, book[i].authorName, book[i].pubYear,
               book[i].isbnNumber);
    }
}

int main() {
    int n;
    printf("Kaç kitap eklemek istiyorsunuz? ");
    scanf("%d", &n);

    // BookDatabase için dinamik bellek ayırma
    BookDatabase *books = (BookDatabase *)malloc(n * sizeof(BookDatabase));

    // Kitapları ekleme
    for (int i = 0; i < n; i++) {
        printf("\n%d. kitap bilgilerini giriniz:\n", i + 1);
        addBook(i + 1, &books[i]);
    }

    // Kitapları görüntüleme
    display(n, books);

    // Belleği serbest bırakma
    free(books);
}
```

```
    return 0;  
}
```

2. Aşağıdaki programın çıktısı nedir? (25 Puan)

```
#include <stdio.h>  
void F(char *str)  
{  
    if(*str=='\0') return;  
    F(str+1);  
    printf("%s\n",str);  
}  
int main()  
{  
    F("Yildiz");  
    return 0;  
}
```

3. "information.txt" adlı bir dosyadan okunan metni alan ve içindeki tüm küçük harfleri büyük harflere dönüştürerek "output.txt" adlı başka bir dosyaya yazan C programını yazınız? (25 Puan)

```
#include <stdio.h>  
  
// Küçük harfi büyük harfe dönüştüren fonksiyon  
char toUpper(char c) {  
    if (c >= 'a' && c <= 'z') {  
        return c - ('a' - 'A');  
    } else {  
        return c;  
    }  
}  
  
int main() {  
    // Giriş dosyası ve çıkış dosyası tanımlamaları  
    FILE *girisDosyasi, *cikisDosyasi;  
    char karakter;  
  
    // Giriş dosyasını açma  
    girisDosyasi = fopen("information.txt", "w+");  
    if (girisDosyasi == NULL) {  
        perror("Hata: Giriş dosyası açılamadı");  
        return 1;  
    }  
  
    // Çıkış dosyasını açma  
    cikisDosyasi = fopen("output.txt", "w");  
    if (cikisDosyasi == NULL) {  
        perror("Hata: Çıkış dosyası açılamadı");  
        return 1;
```

```

}

// Giriş dosyasından karakterleri oku ve büyük harfe dönüştürerek çıkış dosyasına yaz
while ((karakter = fgetc(girisDosyasi)) != EOF) {
    karakter = toupper(karakter);
    fputc(karakter, cikisDosyasi);
}

// Dosyaları kapat
fclose(girisDosyasi);
fclose(cikisDosyasi);

printf("Metin basarıyla donusturuldu ve output.txt dosyasina yazıldı.\n");

return 0;
}

```

4. Aşağıdaki fonksiyonu recursive olarak yeniden yazınız? (25 Puan)

```

int FnR(char *str)
{
    char c;
    while(*str != '\0')
    {
        c = *str;
        if(c > 'A' && c <= 'Z') *str = *str - 1;
        else if(c == 'A') *str = 'Z';
        str++;
    }
    return;
}

```

```

#include <stdio.h>

void FnR(char *str) {
    // Karakter dizisinin sonunu gösteren null karakteri bulduk mu kontrol edelim
    if (*str == '\0') {
        return;
    }

    // Karakteri bir önceki harfe dönüştür
    if (*str > 'A' && *str <= 'Z') {
        *str = *str - 1;
    } else if (*str == 'A') {
        *str = 'Z';
    }

    // Sonraki karakter için recursive çağrı yap
    FnR(str + 1);
}

```

```
int main() {
    char str[] = "HELLO";
    Fnر(str);
    printf("Sonuç: %s\n", str);
    return 0;
}
```