

2024-2025 BAHAR DÖNEMİ

ALGORİTMALAR VE PROGRAMLAMA II

UYGULAMA 9

1. Soru: Dosya Okuma + Hata Yönetimi + errno, strerror, goto, #define

- Kullanıcıdan bir dosya adı alınır.
- Dosya açılır ve içeriği okunur.
- Her aşamada hata kontrolü yapılır.
- goto ile hatalara atlanır.
- errno ve strerror() ile kullanıcıya açıklama yapılır.

Yukarıdaki adımları gerçekleştiren bir C kodu yazınız.

2. Soru: Yapay TRY / CATCH Kullanarak Hata Yönetimi (setjmp / longjmp ile)

- Basit bir matematiksel işlem fonksiyonu içeren,
- Hatalı durumlarda THROW yapan,
- TRY / CATCH makroları ile istisna yönetimini taklit eden

Bir C kodu yazınız.

1. Sorunun çözümü:

```
#include <stdio.h>
#include <errno.h>
#include <string.h>

#define BUFFER_SIZE 256
#define DOSYA_ADI "veriler.txt"

int main() {
    FILE *fp;
    char buffer[BUFFER_SIZE];

    fp = fopen(DOSYA_ADI, "r");

    if (fp == NULL) {
        goto hata;
    }

    printf("Dosya başarıyla açıldı: %s\n", DOSYA_ADI);

    while (fgets(buffer, BUFFER_SIZE, fp)) {
        printf("Okunan satır: %s", buffer);
    }

    fclose(fp);
    return 0;

hata:
    printf("Dosya açılmadı!\n");
    printf("Hata Kodu: %d\n", errno);
    printf("Hata Mesajı: %s\n", strerror(errno));
    return 1;
}
```

2. Sorunun çözümü:

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

jmp_buf hata_yakala;

#define TRY if (setjmp(hata_yakala) == 0)
#define CATCH else
#define THROW longjmp(hata_yakala, 1)

int bolme(int a, int b) {
    if (b == 0) {
        THROW;
    }
    return a / b;
}

int main() {
    int x = 10, y = 0;
    int sonuc;

    TRY {
        printf("İşlem başlatılıyor...\n");
```

```
        sonuc = bolme(x, y);
        printf("Sonuç: %d\n", sonuc);
    }
    CATCH {
        printf("HATA: Sıfıra bölme hatası yakalandı!\n");
    }

    printf("Program sonlandı.\n");
    return 0;
}
```