

Adatbázis Rendszerek I.

BSc

1.gyak.

2021. 09. 15.

Készítette:

Tóth József BProf
Üzem-mérnök-
informatikus alapszak
WI2GDP

1. **feladat** - Építsen fel egy B-fát az alábbi elemekből, melyek beépülési sorrendje adott.

A fa fokszáma 4, és a beszúrandó elemek listája: 6, 12, 9, 2, 5, 4, 15, 20, 1, 3, 10, 14, 17, 16, 21, 25, 24.

```
joseph@joseph-virtual-machine: ~/Documents/WI2GDP/fold
File Edit View Search Terminal Help
joseph@joseph-virtual-machine:~/Documents/WI2GDP$ ls
bokor fa fold
joseph@joseph-virtual-machine:~/Documents/WI2GDP$ cd bokor
joseph@joseph-virtual-machine:~/Documents/WI2GDP/bokor$ mkdir malna
joseph@joseph-virtual-machine:~/Documents/WI2GDP/bokor$ mkdir mogyoro
joseph@joseph-virtual-machine:~/Documents/WI2GDP/bokor$ mkdir szilvia
joseph@joseph-virtual-machine:~/Documents/WI2GDP/bokor$ ls
malna mogyoro szilvia
joseph@joseph-virtual-machine:~/Documents/WI2GDP/bokor$ rm -r szilvia/
joseph@joseph-virtual-machine:~/Documents/WI2GDP/bokor$ ls
malna mogyoro
joseph@joseph-virtual-machine:~/Documents/WI2GDP/bokor$ mkdir szilva
joseph@joseph-virtual-machine:~/Documents/WI2GDP/bokor$ ls
malna mogyoro szilva
joseph@joseph-virtual-machine:~/Documents/WI2GDP/bokor$ cd ..
joseph@joseph-virtual-machine:~/Documents/WI2GDP$ cd fa
joseph@joseph-virtual-machine:~/Documents/WI2GDP/fa$ mkdir alma
joseph@joseph-virtual-machine:~/Documents/WI2GDP/fa$ cd ..
joseph@joseph-virtual-machine:~/Documents/WI2GDP$ cd fol
bash: cd: fol: No such file or directory
joseph@joseph-virtual-machine:~/Documents/WI2GDP$ cd fold
joseph@joseph-virtual-machine:~/Documents/WI2GDP/fold$ mkdir eper
joseph@joseph-virtual-machine:~/Documents/WI2GDP/fold$ mkdir gesztenye
joseph@joseph-virtual-machine:~/Documents/WI2GDP/fold$
```

2. **feladat** - A programozási feladatokat CodeBlocks programmal készítsék el. A projekt neve: XY_neptunkod

A szabvány billentyűzetről olvasson be sorokat, egészen a végjelig (#). A beolvasott sorokat írja ki egy szövegfile-ba (munka.txt). A szövegfile nevét a bevitel első sorában adja meg. Az így létrehozott, lezárt állományt utána nyissa meg és írja vissza a lementett szöveget nagybetűs formában.

```

#include <stdio.h>
#include <stdlib.h>
void pelda();

int main()
{
    pelda();
    return 0;
}

void pelda() {
    FILE *fp;
    char ch;
    char fnev[50];
    printf("Fajlneve (50 karakter): ");
    scanf("%s", fnev);
    fp = fopen(fnev, "w");
    printf("Uzenet: ");
    while((ch=getchar()) != '#')
    {
        putc(ch, fp);
    }
    fclose(fp);

    fp = fopen(fnev, "r");
    while ((ch=getc(fp)) != EOF) {
        printf("%c",ch);
    }
    fclose(fp);
}

```

3. **feladat** - Végezze el egy fájl tartalmának másolását egy másik fájlba (fp1, fp2;).
A fileneveket másoló függvényt hívó részben kell beolvasni.

```

#include <stdio.h>
#include <stdlib.h>
void pelda();

int main()
{
    pelda();
    return 0;
}

void pelda() {
    FILE *fp1, *fp2;
    char ch;
    char fnev[50];
    printf("Olvas fajlneve (50 karakter): ");
    scanf("%s", fnev);
    fp1 = fopen(fnev, "r");

    if(!fp1) {
        printf("Error.");
        exit(0);
    }

    printf("Iras fajlneve (50 karakter): ");
    scanf("%s", fnev);
    fp2 = fopen(fnev, "w");
    while ((ch=getc(fp1)) != EOF) {
        fputc(ch, fp2);
    }
    printf("Done.");
    fclose(fp1);
    fclose(fp2);
}

```

4. **feladat** - Tároljon le auto (rendszer, típus, ár) rekordokat egymás után egy bináris állományban, majd készítsen függvényt az i. rekord visszaolvasására.

```
for(int i = 0; i < db; i++) {
    insert();
}

do {
    printf("Hányadik rekord?");
    scanf("%d", &db);
} while ((c=getchar()) != '\n');
reader(db);

return 0;
}
```

```
void insert() {
    FILE *fp;
    Auto car;
    char fnev[50];

    fp = fopen("munka.txt", "ab");
    if(fp == NULL) {
        printf("Error.\n");
        exit(0);
    }
}
```

```
printf("\nRendszám: ");
scanf("%d", &car.rendszer);
printf("\nTípus: ");
scanf("%s", car.típus);
printf("\nÁr: ");
scanf("%d", &car.ar);
```

```
fwrite(&car, sizeof(Auto), 1, fp);
printf("Írásra kész.");
fclose(fp);
}
```

```
void finder(int fkod) {
    FILE *fp;
    Auto car;
    int i, filesize;
    bool ok = false;
    fp=fopen("munka.txt", "rb");
    if(!fp) {
        printf("Error.");
        return -1;
    }

    fseek(fp, 0L, SEEK_END);
    filesize = ftell(fp)/sizeof(Auto);

    for(i = 0; i < filesize; i++) {
        fseek(fp, sizeof(Auto)*i, SEEK_SET);
        fread(&car, sizeof(Auto), 1, fp);
        if (car.kod == fkod) {
            ok == true;
            printf("\nLétező kód, asszociált adat:");
            printf("\nRendszám: %s, Típus: %s, Ár: %d", car.rendszer, car.típus, car.ar);
        }
    }

    if (ok == false) {
        printf("\nNincs ilyen adat: %d", fkod);
    }

    fclose(fp);
    return 0;
}
```

5. **feladat** - Készítsen programot, mely fel tud vinni személyeket (azonosító és név) bináris fájlba. Készítsen függvényt a) új rekordot létrehozatalára, b) létező rekord törlésére c) létező rekord módosítására

```
typedef struct Auto{
    char rendszam[7];
    char tipus[20];
    int ar;
}Auto;

void readnumber(int* number) {
    char ch;
    bool ok;
    do {
        ok = true;
        if (*number < 0)
        {
            printf("Hibas input\n");
            ok = false;
        }
    } while(!ok);
}

int main()
{
    int db;
    char c;
    do {
        printf("Mennyi rekord?");
        scanf("%d", &db);
    } while ((c=getchar()) != '\n' && c != EOF);

    for(int i = 0; i < db; i++) {
        insert();
    }

    do {
        printf("Hanyadik rekord?");
        scanf("%d", &db);
    } while ((c=getchar()) != '\n');
    finder(db);
}
```

6. **feladat** - Az autókat tároló adatfile-ban végezze el az alábbi lekérdezési műveleteket:

- Számítsa ki a fájlban eltárolt autók átlagárát.
- Kérdezze le az eltárolt piros autók darabszámát.
- Keresse meg a legdrágább autót a fájlban.

