

Adatbázis Rendszerek I. BSc

1.gyak.

2021.09.15

Készítette:

Orosz Dániel Bsc

Üzemmérnök-informatikus

C5S7FM

Miskolc, 2021

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.

2. feladat

A programozási feladatokat CodeBlocks programmal készítsék el.

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("File name (50 characters): ");
    scanf("%s", fnev);
    fp = fopen(fnev, "w");
    printf("Message: ");
    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("Olvasas 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){
        printf(ch, fp2);
    }
    printf("Kesz.");
    fclose(fp1);
    fclose(fp2);
}
```

4. feladat

Tároljon le auto (rendszám, típus, ár) rekordokat egymás után egy bináris állományban, majd készítsen függvényt az i. rekord visszaolvasására.

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

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

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

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

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

    return 0;
}

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

    fp = fopen("munka.txt", "ab");

    printf("\nTípus: ");
    scanf("%s", car.tipus);
    printf("\nÁr: ");
    scanf("%d", &car.ar);

    fwrite(&car, sizeof(Auto), 1, fp);
    printf("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.rendszam == fkod) {
            ok = true;
            printf("\nLetezo kod.");
            printf("\nRendszan: %s, Tipus: %s, Ara: %d", car.rendszam, car.tipus, &car.ar);
        }
    }
    if (ok == false) {
        printf("\nNem letezo kod: %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

Útmutató:

- használjon bináris file-t
- használjon struct szerkezetet
- használja az fseek(), ftell() függvényeket
- törlésnél másolja át a maradó részt

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

void insert();
int finder(int fkd);
void readnumber(int* number);

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("Hany elem?");
        scanf("%d", &db);
    } while ((c=getchar()) != '\n' && c != EOF);

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

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

    return 0;
}

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

    fp = fopen("munka.txt", "ab");
    if(fp == NULL) {
        printf("Error.\n");
        exit(0);
    }
    do {
        printf("\nRendszam: ");
        scanf("%d", &car.rendszam);
        printf("\nTipus: ");
        scanf("%[^\\n]s", &car.tipus);
        printf("\nAr: ");
        scanf("%d", &car.ar);
    } while((ch=getchar()) != '\n');

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

```

    printf("Kesz.");
    fclose(fp);
}

int 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 (i == fkod) {
            printf("\nLetezo kod:");
            printf("\nRendszam: %s, Tipus: %s, Ara: %d", car.rendszam, car.tipus, car.ar);
        }
    }
    if (ok == false) {
        printf("\nNem letezo kod: %d", fkod);
    }
    fclose(fp);
    return 0;
}

```

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.

Útmutató:

- olvassa át az állományt rekordonként
- a szükséges adatokat emelje ki a rekordból
- végezze ez a szükséges számításokat memória változók segítségével.

```
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("Hany elem?");
        scanf("%d", &db);
    } while ((c=getchar()) != '\n' && c != EOF);

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

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

    return 0;
}
```