

```

#include <stdio.h>
#include <stdlib.h>
#include <locale.h> //területi beállításhoz setlocale

int main()
{
    system("chcp 1250 > nul"); //windowson működik ékezetekhez
    setlocale(LC_ALL, ""); //Területi beállítás Linuxon is működik ékezetekhez
    //deklaráljuk és literáljuk a változókat
    //literáljuk értéket adunk neki
    int minutesEntered = 0;
    double years = 0.0; //évek
    double days = 0.0; //napok
    double minutesInYear = 0; //percek egy évben
    // get input from user
    //bekérjük az adatokat
    printf("Please enter the number of minutes: ");
    //elmentjük a bekért adatot
    scanf("%d", &minutesEntered);

    minutesInYear = (60 * 24 * 365);

    years = (minutesEntered / minutesInYear);
    days = ((minutesEntered / 60.0) / 24);

    printf("%d minutes in approximately %f years and %f days:
\n", minutesEntered, years, days );
    //most magyarul 2 tizedesig %.2f
    printf("%d perc körülbelül %.2f évnek és %.2f napnak felel meg.\n",
minutesEntered, years, days);

    return 0;
}

```

Természetesen, Ati! Itt egy részletes, lépésről lépésre történő magyarázat a programodhoz **magyarul**, hogy teljesen átlásd, mit csinál:

1. Fejlécfájlok (#include)

```

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

```

- `stdio.h`: A standard input/output függvényeket tartalmazza, például `printf`, `scanf`.
- `stdlib.h`: Általános függvények (pl. `system()`), memórafoglalás, átalakítás, kilépés stb.
- `locale.h`: A `setlocale()` függvényhez kell, amely lehetővé teszi a magyar ékezetes karakterek kezelését.

2. Kódlap és területi beállítás

```
system("chcp 1250 > nul");  
setlocale(LC_ALL, "");
```

- `system("chcp 1250 > nul");`
 - **Windows-specifikus** parancs: beállítja a **1250-es (Közép-európai)** kódlapot, hogy az ékezetek (pl. á, é, ő) rendesen megjelenjenek.
 - A `> nul` elnyeli a kimenetet, hogy ne írjon felesleges szöveget a képernyőre.
- `setlocale(LC_ALL, "");`
 - A rendszer nyelvi/területi beállításait használja (például dátum, karakterkódolás, ékezetek).
 - Linuxon és Windowson is működik.

3. Változók deklarálása és inicializálása

```
int minutesEntered = 0;  
double years = 0.0;  
double days = 0.0;  
double minutesInYear = 0;
```

- `minutesEntered`: A felhasználótól bekért percek számát tárolja.
- `years`: A percekből számolt évek (pl. 525600 perc = kb. 1 év).
- `days`: A percekből számolt napok.
- `minutesInYear`: Egy évben lévő percek száma (később kiszámoljuk: $60 \times 24 \times 365 = 525600$).

4. Felhasználói adatbekérés

```
printf("Please enter the number of minutes: ");  
scanf("%d", &minutesEntered);
```

- Kiírja: **"Please enter the number of minutes:"** = „Kérem adja meg a percek számát:”
- `scanf()` beolvassa az egész számot a `minutesEntered` változóba.

5. Számítások

```
minutesInYear = (60 * 24 * 365);  
years = (minutesEntered / minutesInYear);  
days = ((minutesEntered / 60.0) / 24);
```

- `minutesInYear`: Egy év hány percből áll:
 - 1 óra = 60 perc
 - 1 nap = 24 óra $\rightarrow 60 \times 24 = 1440$ perc
 - 1 év = 365 nap $\rightarrow 1440 \times 365 = 525600$ perc
- `years`: Hány évnyi időt jelent az adott perc:

- például: 1051200 perc → kb. 2 év
 - `days`: Hány napot jelent a megadott perc:
 - `(minutes / 60)` → órák száma
 - osztva 24-gyel → napok száma
-

6. Eredmény kiírása

```
printf("%d minutes in approximately %f years and %f days: \n", minutesEntered,
years , days);
printf("%d perc körülbelül %.2f évnek és %.2f napnak felel meg.\n",
minutesEntered, years, days);
```

- **Első sor:** angolul írja ki az eredményt.
 - **Második sor:** magyarul, **2 tizedesjegyre kerekítve**.
-

7. Visszatérés

```
return 0;
```

- A `main()` program sikeres lefutását jelzi.

Példa futásra:

Bemenet:

```
Please enter the number of minutes: 1000000
```

Kimenet:

```
1000000 minutes in approximately 1.902587 years and 694.444458 days:
1000000 perc körülbelül 1.90 évnek és 694.44 napnak felel meg.
```
