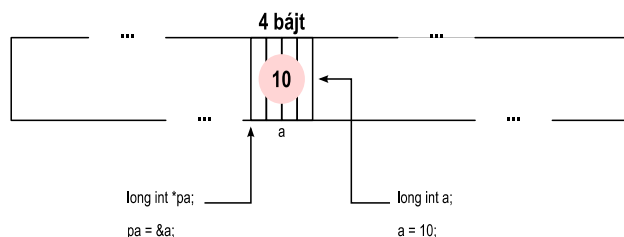


## 12-13. ÓRA

## Változók további tulajdonságai (jellemzői)

- eddig jellemzők: név, típus, érték, érvényesség, jelleg, méret
- újabb jellemző: **cím** => memóriában való pontos elhelyezkedése

## Változó értékének és címének értelmezése



- 'pa' változó egy mutató típusú változó, amelynek értéke egy konkrét memória blokk (bájt) címe
- az 'a' hivatkozás az 'a' változó ÉRTÉKÉT jelöli, ami a példában 10
- az '&a' hivatkozás az 'a' változó memóriában elfoglalt helyének CÍMÉT jelöli, ami annyi amennyi :-)
- az 'a' változó értéke az alábbi megoldásokkal változtatható
  - a = 10 (ez a szokásos forma, eddig ezt használtuk)
  - \*pa = 10 (ez új! a változó értékét úgy változtatjuk meg, hogy a címe alapján elért 4 bájt értékét változtatjuk)
  - az pedig a pa = &a értékadás miatt éppen az 'a' változó címe!
- vagyis egy változó címét úgy kapjuk meg, hogy a változó neve elé leírjuk az '&' jelet

## Alapvető I/O műveletek

- korábban már említettük, hogy a C alapvető függvénykönyvtárát az **stdlib** gyűjteményt
- ennek az **stdio.h** gyűjteménye tartalmazza az alapvető adatbeviteli és –kiviteli szolgáltatásokat (I/O műveletek)
- ezek közül a **printf** függvényt már többször használtuk

- feladata, hogy az alapértelmezett kimeneten (*stdout*) formázott módon adatokat jelenítsen meg
- a függvény legalább egy paramétert vár, ami egy formátum leíró szöveg
- ez egy idézőjelek (") közötti karaktersorozat, amelyben vezérlőkarakterek is lehetnek
- szintaxis (használatának) szabálya: `int printf(const char *format, ...);`
- a '...' jelentése: további paraméterek is megadhatók a függvény számára

```
– printf("\nÁtlag: %6.2f", atlag);
```

- `printf` => kiíró *stdio* függvény neve
- `"\nÁtlag: %6.2f"` => formátum leíró szöveg (sztring)
- `\n` => 'új sor' karakter (ASCII tábla szerinti értéke 10)
- `Átlag:` => megjelenítendő (konstans) szöveg
- `%` => formátum leírás vezérlőkaraktere
- `6.2` => változó kijelzés szélesség és pontosság értékei (6 jegyből 2 tizedes)
- `f` => (float) lebegőpontos érték kijelzését írja elő
- `atlag` => egy átlagos értéket tároló változó **értéke** (véltetően valós szám)

- `printf` típuskijelző kódja (konverziós karakterei)

- `%c` => karakter (egyetlen darab)
- `%d, %i` => egész szám (int)
- `%f` => lebegőpontos szám (float)
- `%s` => sztring (karakter sorozat)

- **scanf** adatbeolvasó függvény

- feladata, hogy az alapértelmezett bemenetről (*stdin*) típusos adatokat olvasson be
- a `printf` függvénynek megfelelő működés, de itt nem a változó értéke, hanem címe a további paraméter
- szintaxis (használatának) szabálya: `int scanf(const char *format, ...);`

```
– scanf("%d", &darab);
```

- `scanf` => beolvasó *stdio* függvény neve
- `"%d"` => formátum leíró szöveg (sztring)
- `%` => formátum leírás vezérlőkaraktere
- `d` => (int) egész érték beolvasását írja elő
- `&darab` => egy darabszámot tároló változó **címe** (véltetően egész szám)

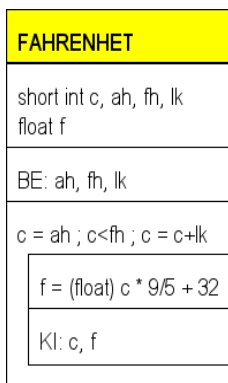
## Feladat

Készítsen programot, amely tetszőleges °C tarmányban tetszőleges °C-onként kiszámítja a F értékeket és táblázatszerű formában kijelzi mind a °C, mind a F értékeket a kijelzőn! A tetszőleges értékeket (alsó határ, felső határ, lépésköz) a scanf függvény segítségével a felhasználótól kérje be! Készítsen struktogramot a probléma megoldására!

### Gondolatok a megoldáshoz

- az alsóhatár és a felsőhatár "emberi" körülmények között -50 °C és +50 °C között lehet
- az értékek tárolásához rövid egész típus (2 bájtt) bőven elegendő
- a lépésköz tárolásához ennek megfelelően szintén elegendő a rövid egész típus
- a Fahrenheit érték tárolására továbbra is valós típust alkalmazunk

### Struktogram készítése



### Forrásprogram készítése

```
#include <stdio.h>

int main(void) {
    short int c, ah, fh, lk;
    float f;

    printf("\nAdja meg a következő adatokat:\n");

    printf("- alsó határ... " );
    scanf("%d", &ah);

    printf("- felső határ... " );
    scanf("%d", &fh);

    printf("- lépésköz..... " );
    scanf("%d", &lk);

    for (c=ah; c<fh; c = c+lk) {
        f = (float) c*9/5+32;
        printf("\n%5d°C | %9.2fF", c, f);
    }

    return (0);
}
```

### Program futtatása

- a felhasználói adatok megadása után a program látszólag semmit sem tesz
- vagyis a programban valamilyen SZEMANTIKAI hiba van
- az ilyen hibák megkeresésére használhatjuk az integrált környezetek DEBUG szolgáltatását
- ilyen szolgáltatás azonban most nem áll rendelkezésünkre

### Mi lehet a program hibája? :-)

### Probléma kezelése

- a "%d" formátum egész számra (4 bájtt) érvényes, vagyis a scanf 4 bájtt értékét változtatja meg a memóriában
- az 'ah' változónk azonban rövid egész (short int) típusú, ami csak 2 bájtt nagyságú
- **vagyis a scanf függvény fontos memóriaterületet ír felül!**

### Házi feladat

- scanf függvény alkalmazási hibájának megoldása