

```

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

int main()
{
    unsigned long long sum = 0LL;//0LL
    unsigned int count = 0;

    //Read the number of integers to be summed
    printf("\nEnter the number of integers you want to sum: ");
    scanf(" %u", &count);
    //Sum integers from 1 to count
    unsigned int loop;
    for (loop = 1; loop <= count ; ++loop)
    {
        printf("inside loop");
        sum += loop;
    }

    printf("\nTotal of the first %u numbers is %llu\n", count, sum);

    return 0;
}

```

Ez a C program egy **egyszerű számláló és összegző program**, ami a megadott számig (1-től `count`-ig) **összeadja az egész számokat**.

1. Változók:

```
unsigned long long sum = 0LL;
```

- `sum`: az összeget tárolja (kezdetben 0), `unsigned long long` típus, hogy **nagy számokat** is tudjon tárolni.
- `0LL` azt jelenti: "0" long long típusban.

```
unsigned int count = 0;
```

- `count`: a felhasználótól bekért szám, eddig fogunk összeadni.

2. Bemenet:

```
printf("\nEnter the number of integers you want to sum: ");
scanf(" %u", &count);
```

- Bekéri a felhasználótól, hogy hány számot szeretne összegezni (`count` értéke).

3. Ciklus – összeadás:

```
for (loop = 1; loop <= count; ++loop)
{
    printf("inside loop");
    sum += loop;
}
```

- `loop` 1-től `count`-ig megy, minden lépésben hozzáadja `loop` értékét a `sum`-hoz.
- Közben kiírja minden iterációban: `"inside loop"` (nem túl hasznos így, de talán debuggolásra szánták).

Példa: ha `count = 5`, akkor:

- `sum = 1 + 2 + 3 + 4 + 5 = 15`

4. Eredmény kiírása:

```
printf("\nTotal of the first %u numbers is %lu\n", count, sum);
```

- Kiírja az összeget (`sum`) és a darabszámot (`count`).

Összefoglalva:

A program:

1. Megkérdezi, hány számot szeretnél összegezni.
 2. Összeadja az 1-től `count`-ig terjedő számokat.
 3. Kiírja az eredményt.
-