Feladat 1

Írjon egy **int megfordit(int num)** függvényt, amely az input vektorban megadott num (max 10 számjegy) számjegyeit megfordítja.

Példa: Ha a **bemenet:** 47323 **kimenet:** 32374

```
int megfordit(int num)
{
    int res = 0;
    while (num != 0) {
        int n = num % 10;
        res = res * 10 + n;
        num /= 10;
    }
    return res;
}
```

Feladat 2

Írjon egy **int megfordit(int num)** függvényt, amely az input vektorban megadott num bitjeit megfordítja.

```
int megfordit(int num)
{
    return ~res;
}
```

Feladat 3

Írjon egy **void binaris(short num)** függvényt, amely a num számot bináris formában írja ki.

```
void binaris(short num)
{
    short pos;
    for (pos = (sizeof(short) * 8 -1) ; pos >= 0 ; pos--)
    {
      (num & (1 << pos))? printf("1"): printf("0");
    }
}
```

Feladat 4

Írjon egy **short atbillent(short num)** függvényt, amely a num szám második bitjét az ellenkezőjére

változtatja, majd visszaadja az így kapott számot.

Példa: Ha a **bemenet:** 1 **kimenet:** 3

```
short atbillent(short num)
{
    return num ^ 0b10;
}
```

Feladat 5

Írjon egy **short atbillent(short num)** függvényt, amely a num szám második és negyedik bitjét az ellenkezőjére változtatja, majd visszaadja az így kapott számot.

Példa: Ha a **bemenet:** 1 **kimenet:** 3

```
short atbillent(short num)
{
    return num ^ 0b1010;
}
```

Feladat 6

Írjon egy **int bitstatusz(int num)** függvényt, amely a num szám harmadik bitjének státuszát adja vissza.

Példa: Ha a **bemenet:** 5 **kimenet:** 1

```
int bitstatusz(int num)
{
    return (num >> 2) & 1;
}
```

Feladat 7

Írjon egy **int bithossz(int num)** függvényt, amely a num szám bináris értékének tényleges hosszát adja vissza.

Példa: Ha a **bemenet:** 5 (101) **kimenet:** 3

```
int bithossz(int num)
{
    int count = 0, i;
    if(num == 0) return 0;
```

```
    for(i=0; i< 32; i++) {
        if((1 << i) & num) count = i;
    }
    return ++count;
}
```

Feladat 8

Írjon egy **int egyesbitek(int num)** függvényt, amely a num szám 1-es bitjeit számolja meg, majd visszaadja ezt az értéket.

Példa: Ha a **bemenet:** 5 (101) **kimenet:** 2

```
int egyesbitek(int num) {
    int i, count=0;
    for(i=0; i< sizeof(int)*8; i++) {
        if( num & (1<< i) )
            count++;
    }
    return count;
}
```

Feladat 9

Írjon egy **float matrixatlag(int sor, int oszlop, int matrix[sor][oszlop])** függvényt, amely átveszi egy mátrix méreteit és a mátrixot, majd visszaadja az átlagát.

```
float matrixatlag(int sor, int oszlop, int matrix[sor][oszlop]) {
    long sum = 0;
    for(int i=0; i < sor; i++)
        for(int j=0; j < oszlop; j++)
            sum += matrix[i][j];
    return (float)sum/(sor*oszlop);
}
```

Feladat 10

Írjon egy **int csere(int num)** függvényt, amely a num szám számjegyei közül az összes 0-t 1-re cseréli.

```
int csere(int num)
{
    if (num == 0) return 0;
    int digit = num % 10;
    if (digit == 0)
```

```
        digit = 1;
    return csere(num/10) * 10 + digit;
}
```

From:
https://edu.iit.uni-miskolc.hu/ - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:szamitastechnika:teszt_feladatok_5

Last update: **2025/09/18 12:39**