1. feladat: Milyen hibákat talál az alábbi megoldásokban?

```
m = malloc(80);
m = NULL;
```

megoldás: 80 bájt lefoglalása, majd a pointer nullázása. Probléma: - a lefoglalt 80 byte nem szabadul fel.

```
free(n);
n = 5;
```

megoldás: felszabadított memóriaterületre akarunk írni.

```
char *p;
*p = 'a';
```

megoldás: nem inicializált pointert akarunk használni.

2.Feladat:: Az alábbi 3 függvényből melyik helyes, illetve hibás?

```
int *f(void)
{
  int x = 10;
  return (&x);
}
int *g(void)
{
  int * p;
  *p = 10;
  return p;
}
int *h(void)
{
  int *p;
  p = (int *) malloc (sizeof(int));
  *p = 10;
  return p;
}
```

megoldás: csak a h() függvény helyes. A f() és g() hibásak, mert ezekben az esetekben a memóriacím a veremben jön létre és a visszaadott érték helytelen memóriacímre fog mutatni.

3.Feladat:: Mi lesz az alábbi program kimenete?

```
#include<stdio.h>

void f(int *a)
{
    a = (int*)malloc(sizeof(int));
}

int main()
{
    int *p;
    f(p);
    *p = 10;
    printf("%d",*p);
}
```

megoldás: nem ír ki semmit, hanem lefagy.

4.Feladat:: Hogyan lehetne kijavítani az előző feladatban szereplő programot?

megoldás: az előző program azért nem működik, mert a pointer nincs inicializálva, ezért a 0-s memóriacímet adja át az f() függvénynek, így a malloc lefoglalja a 4 byte-ot, de hibás helyre írja vissza. A javítás során vegyük figyelembe, hogy egy pointerre mutató pointer már ténylegesen a p változó címét adja vissza, amibe már a f() malloc()-ja már be tudja írni a lefoglalt memória címét.

```
#include<stdio.h>

void f(int **a)
{
    *a = (int*)malloc(sizeof(int));
}

int main()
{
    int *p;
    f(&p);
    *p = 10;
    printf("%d",*p);
}
```

5. Feladat:: Mi a probléma következő programmal?

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

```
int main()
{
    float *p = (float *)malloc(sizeof(float));
    p = NULL;

    free(p);
}
```

megoldás: a p NULL-ázása után, a free() nem tudja, hogy hol van az a memóriacím, amit fel kell szabadítani. A free() nem a p változót, hanem az abban tárolt címet szabadítja fel.

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:memoria_kezeles_peldak

Last update: **2025/11/25 19:46**