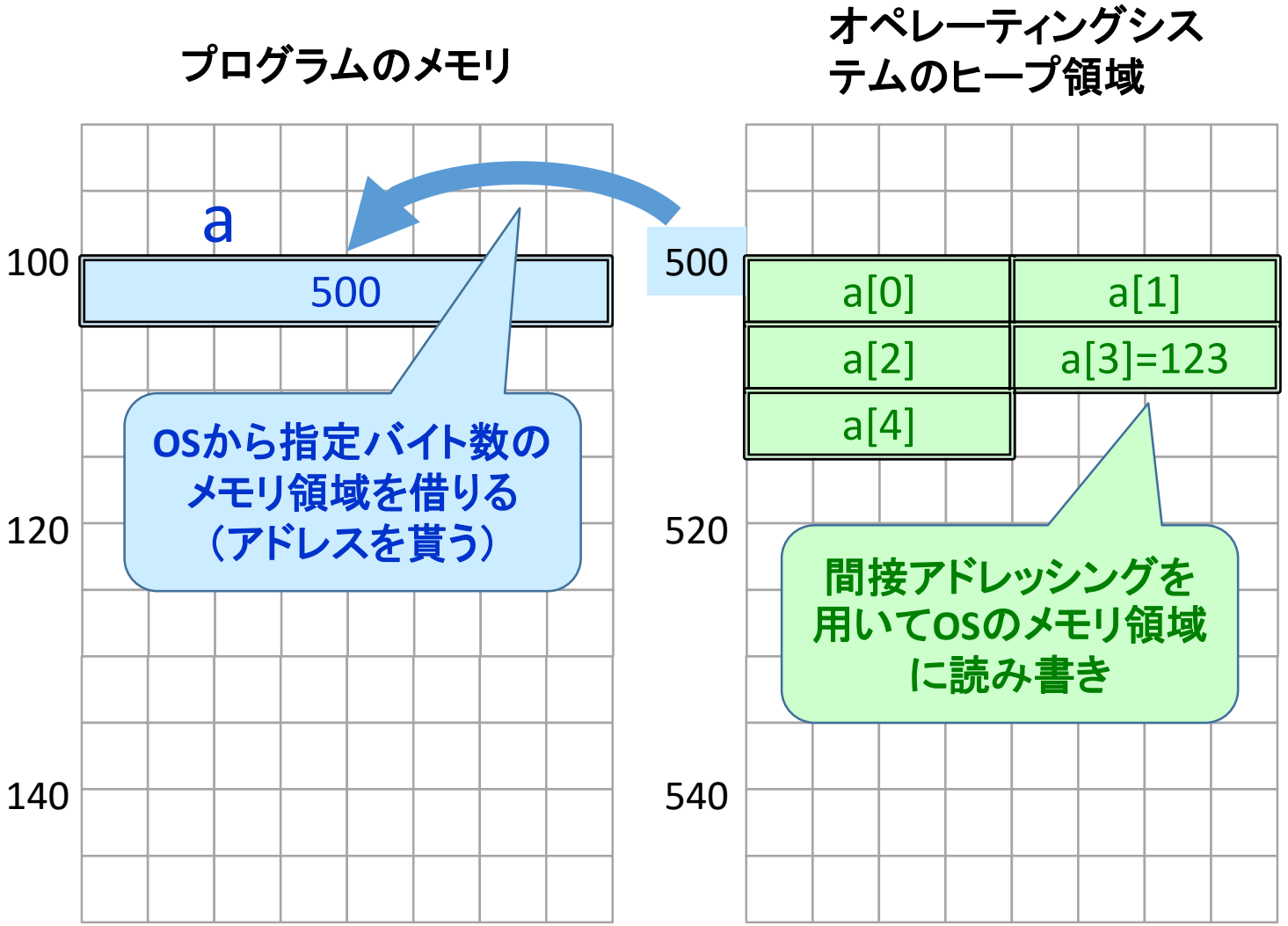
[B5]
動的割り当て

メモリ領域の動的割り当て

- 使用するメモリ領域を、プログラムの実行中にOSから借りたり返却したりできる
- (アドレス)=(型 *)malloc(Byte数);
  - **OSから指定バイト数のメモリ領域を借りる**
  - **使用できるメモリ領域のアドレスを貰う**
- a[3]=123;
  - **間接アドレッシングを用いてOSのメモリ領域にアクセス**
  - **普通の配列と同じように読み書きできる**
- free(アドレス);
  - **借りたメモリ領域は使い終わったら「必ず」返す**

malloc()

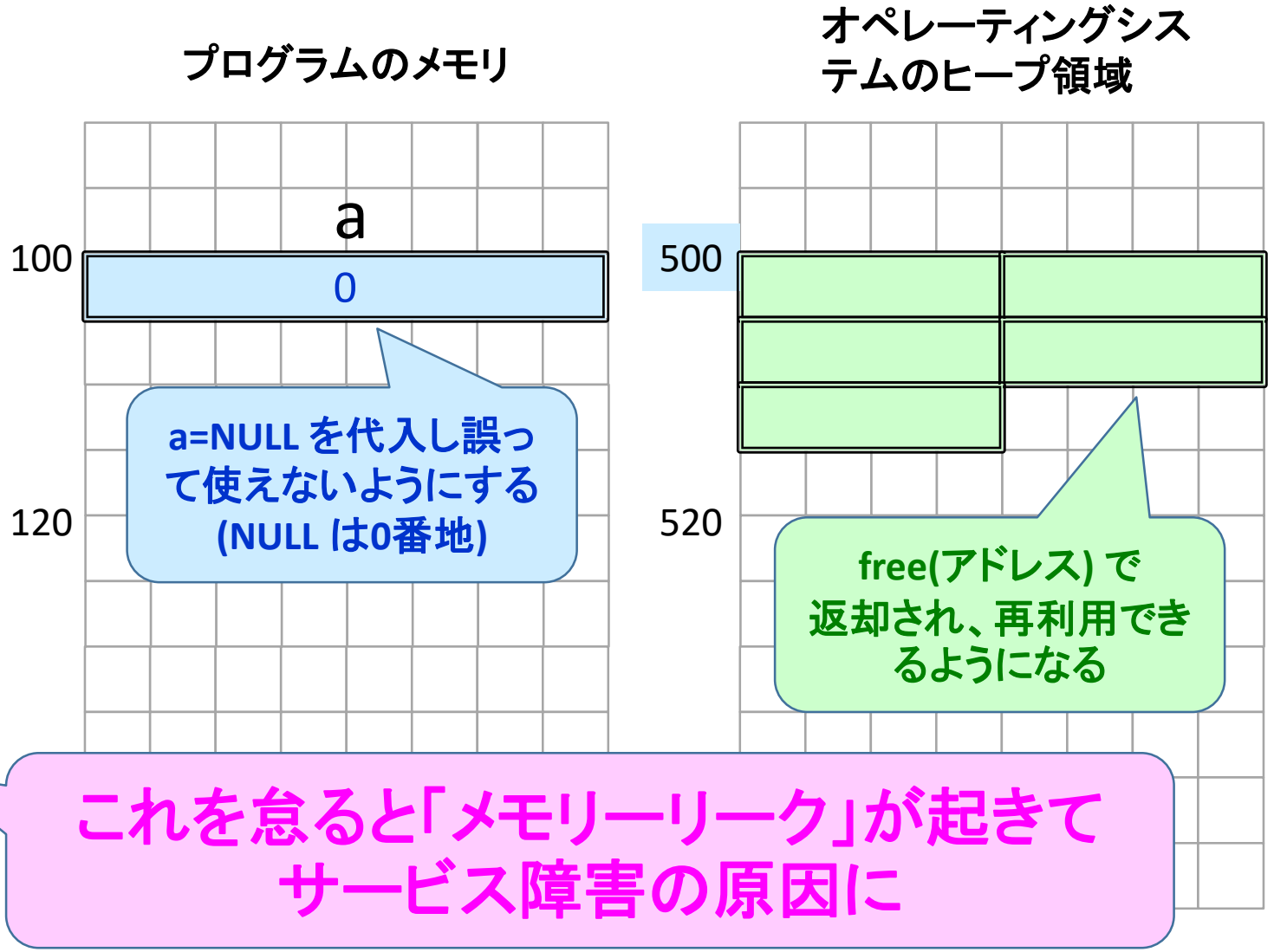
```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
  int  *a;

  a = (int *)malloc(sizeof(int)*5);

  a[3] = 123;

  free(a);
  a = NULL;

  return 0;
}
```



free()

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
  int  *a;

  a = (int *)malloc(sizeof(int)*5);

  a[3] = 123;

  free(a);
  a = NULL;

  return 0;
}
```

借りたメモリ領域は使い終わったら「必ず」返す

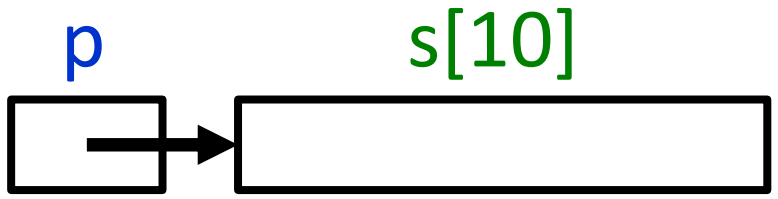


これを怠ると「メモリーリーク」が起きてサービス障害の原因に

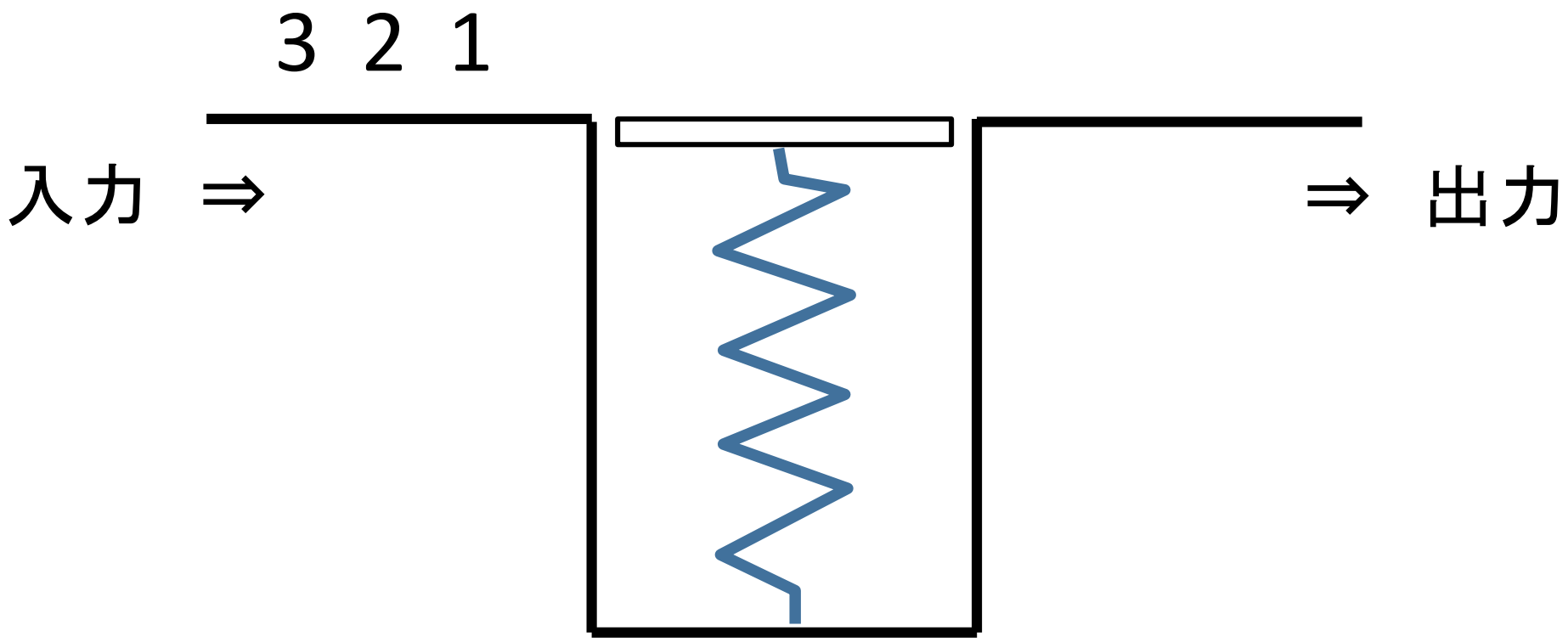
ポインタと矢印表現

```
#include <stdio.h>
#include <string.h>
int main(void)
{
  char *p,s[10];
  p = s;
  strcpy(s,” Hello” );
  return 0;
}
```

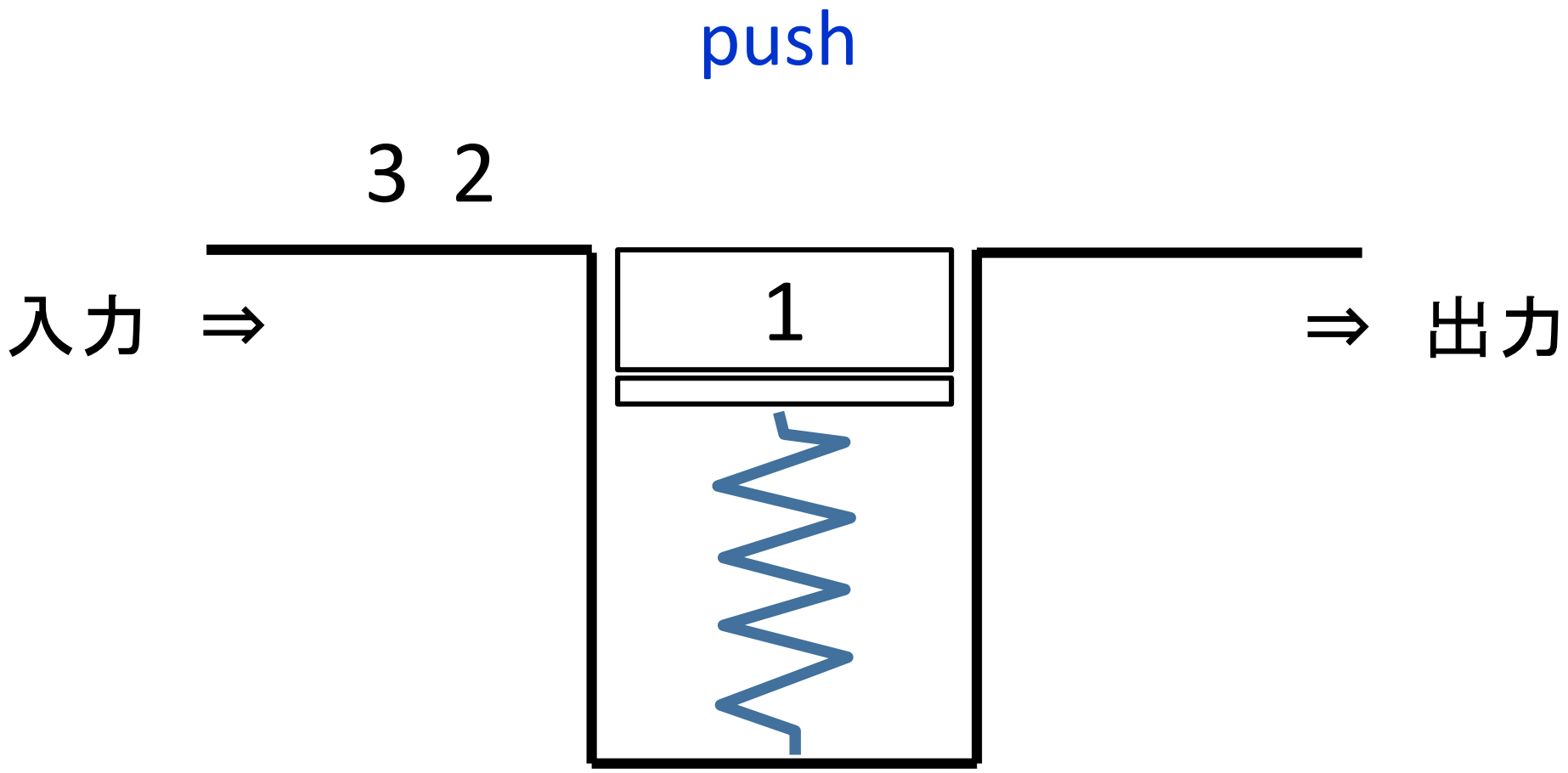
ポインタ p に配列 s[ ] のアドレスが
格納されている状態を、
右図のように矢印で表現する



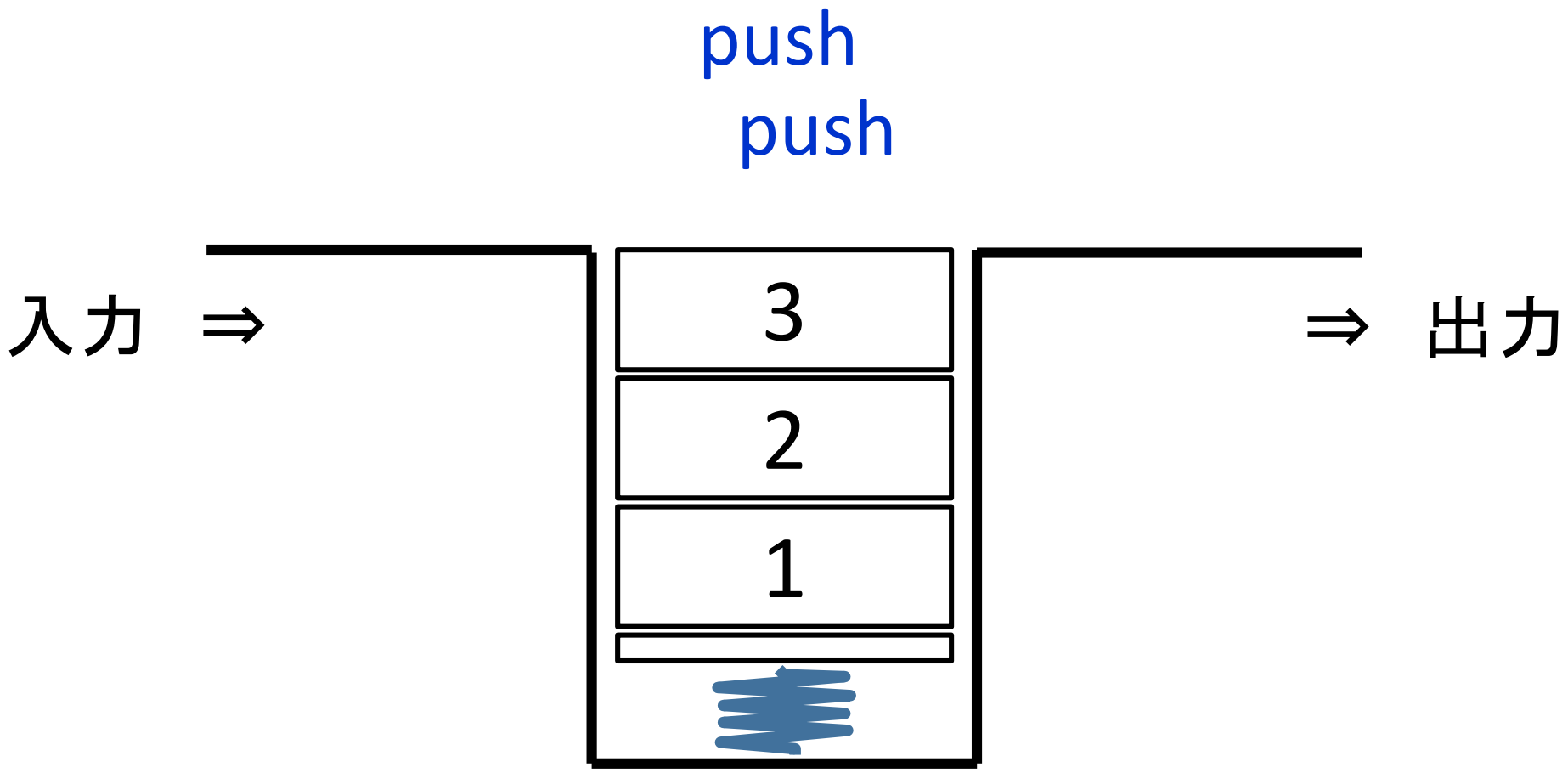
データ構造「スタック」



データ構造「スタック」

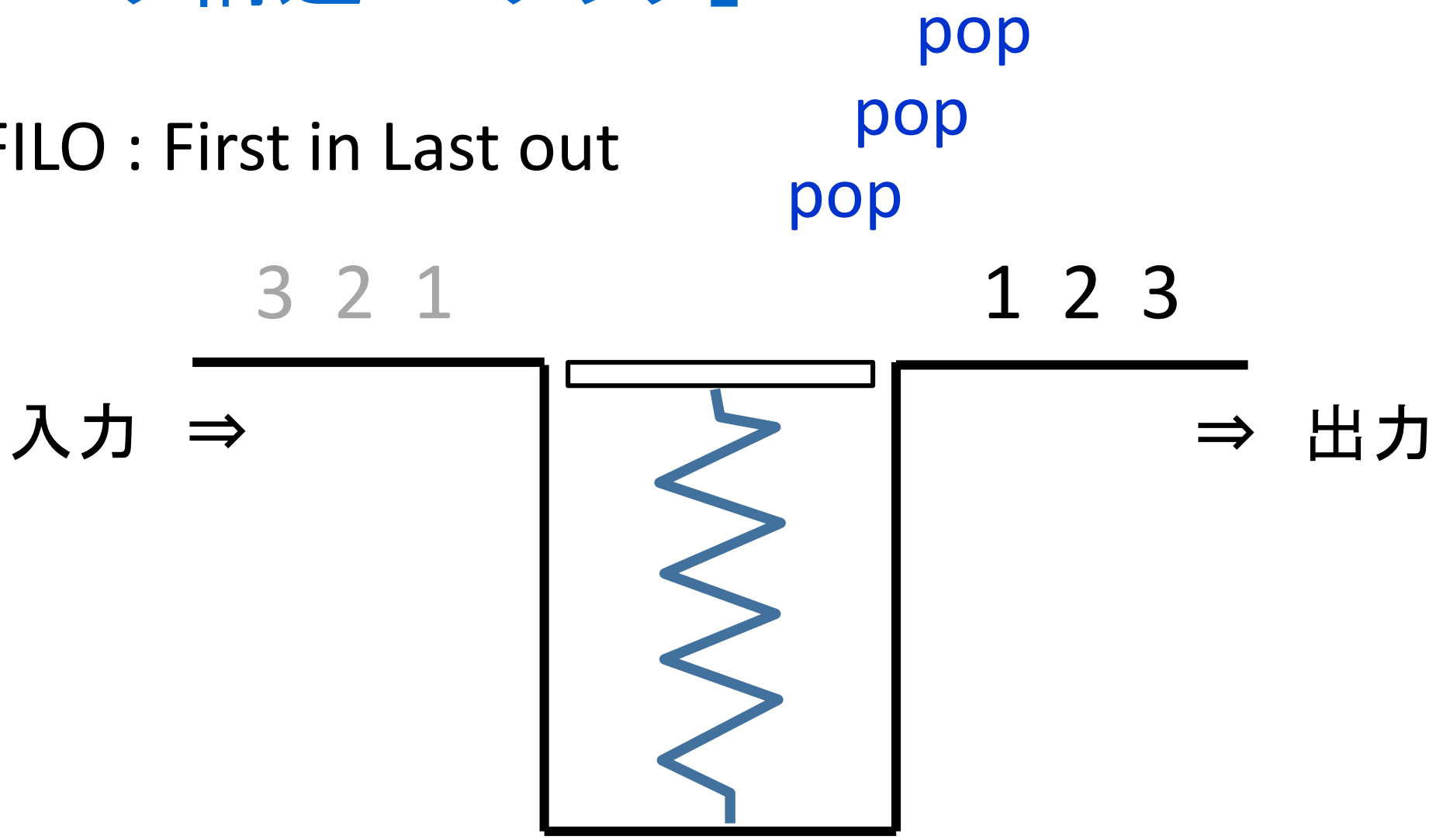


データ構造「スタック」



データ構造「スタック」

- FILO : First in Last out



スタックとメモリ領域の動的割り当て

- スタックの深さ(サイズ)を実行時に指定したい
- istack_t *istack_new(int size);
  - この中で malloc() し、スタック構造体とデータ領域のメモリを確保
- void istack_delete(istack_t *);
  - この中で free() し、データ領域とスタック構造体のメモリを解放 （順番に注意）