

► 动态数组作函数参数

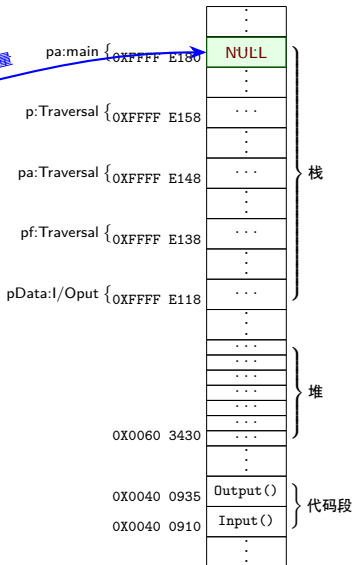
```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++){
        pf(p); //调用传入的函数
    }
}

void Input(int * pData)
{
    scanf("%d", pData);
}

void Output(int * pData)
{
    printf("%d ", *pData);
}
```

声明指针变量



► 动态数组作函数参数

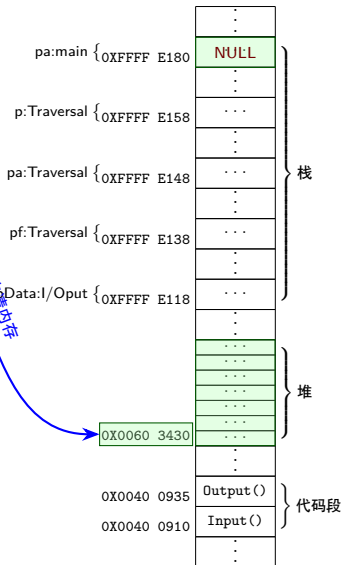
```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++){
        pf(p); //调用传入的函数
    }
}

void Input(int * pData)
{
    scanf("%d", pData);
}

void Output(int * pData)
{
    printf("%d ", *pData);
}
```

动态申请内存



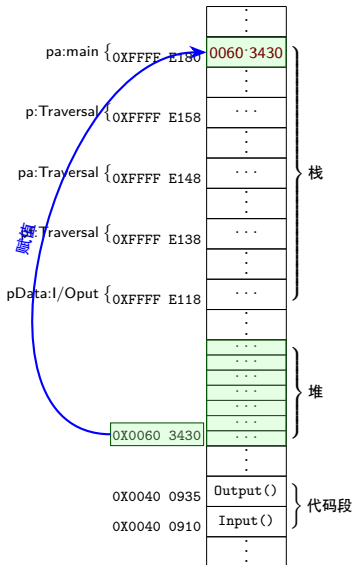
► 动态数组作函数参数

```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++){
        pf(p); //调用传入的函数
    }
}

void Input(int * pData)
{
    scanf("%d", pData);
}

void Output(int * pData)
{
    printf("%d ", *pData);
}
```



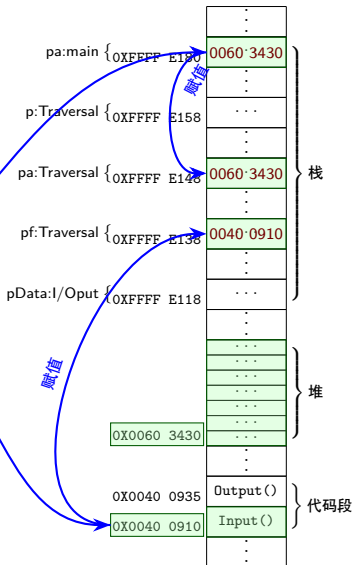
▶ 动态数组作函数参数

```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++){
        pf(p); //调用传入的函数
    }
}

void Input(int * pData)
{
    scanf("%d", pData);
}

void Output(int * pData)
{
    printf("%d ", *pData);
}
}
```



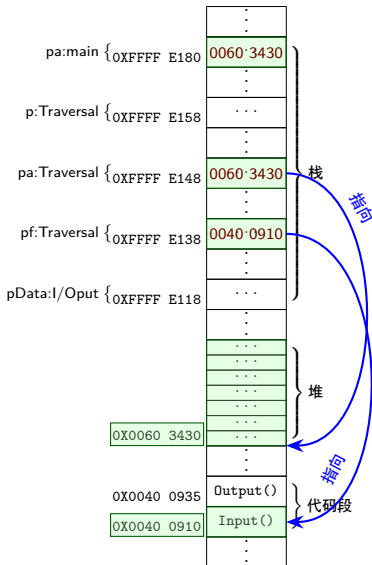
► 动态数组作函数参数

```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++){
        pf(p); //调用传入的函数
    }
}

void Input(int * pData)
{
    scanf("%d", pData);
}

void Output(int * pData)
{
    printf("%d ", *pData);
}
```



► 动态数组作函数参数

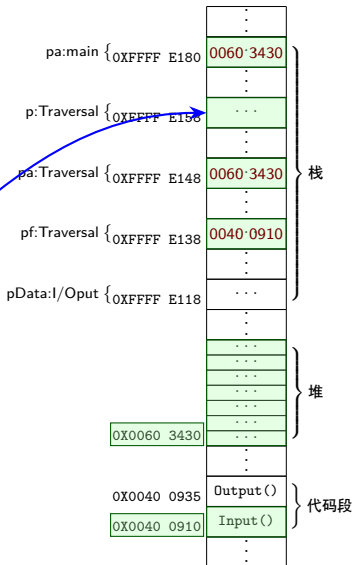
```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++){
        pf(p); //调用传入的函数
    }
}

void Input(int *pData)
{
    scanf("%d", pData);
}

void Output(int *pData)
{
    printf("%d ", *pData);
}
```

声明指针变量



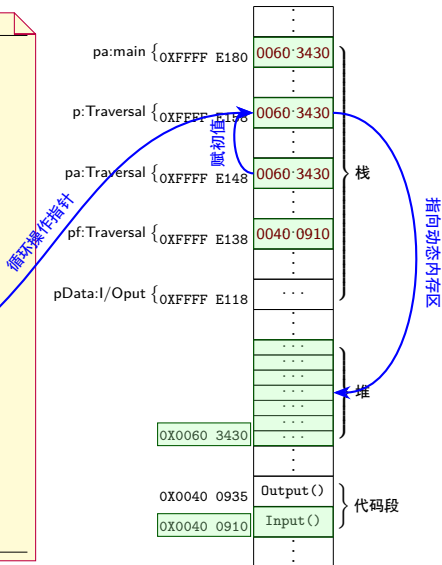
► 动态数组作函数参数

```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++) {
        pf(p); //调用传入的函数
    }
}

void Input(int *pData)
{
    scanf("%d", pData);
}

void Output(int *pData)
{
    printf("%d ", *pData);
}
```



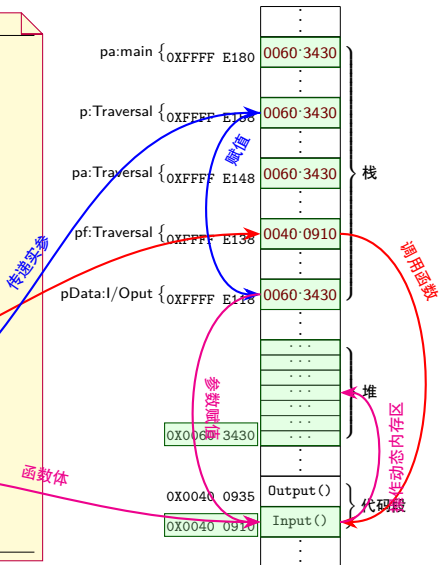
► 动态数组作函数参数

```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++) {
        pf(p); //调用子函数
    }
}

void Input(int *pData)
{
    scanf("%d", pData);
}

void Output(int *pData)
{
    printf("%d ", *pData);
}
```



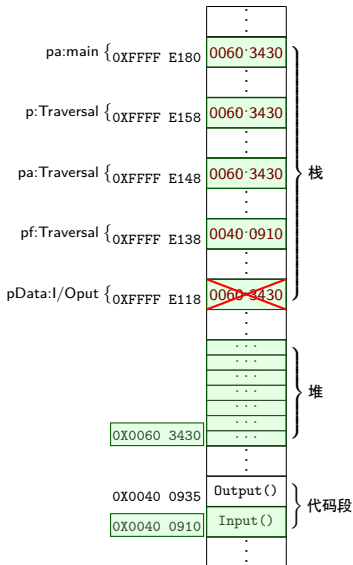
► 动态数组作函数参数

```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++){
        pf(p); //调用传入的函数
    }
}

void Input(int *pData)
{
    scanf("%d", pData);
}

void Output(int *pData)
{
    printf("%d ", *pData);
}
```



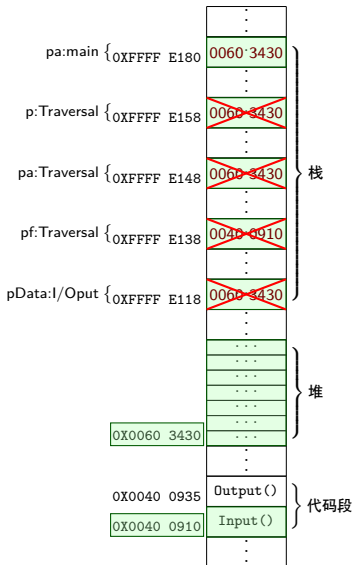
► 动态数组作函数参数

```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++){
        pf(p); //调用传入的函数
    }
}

void Input(int * pData)
{
    scanf("%d", pData);
}

void Output(int * pData)
{
    printf("%d ", *pData);
}
```



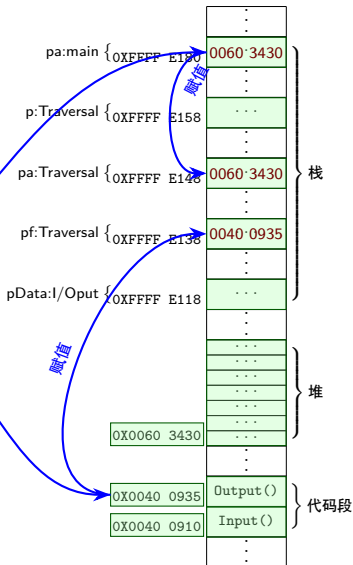
► 动态数组作函数参数

```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++){
        pf(p); //调用传入的函数
    }
}

void Input(int * pData)
{
    scanf("%d", pData);
}

void Output(int * pData)
{
    printf("%d ", *pData);
}
```



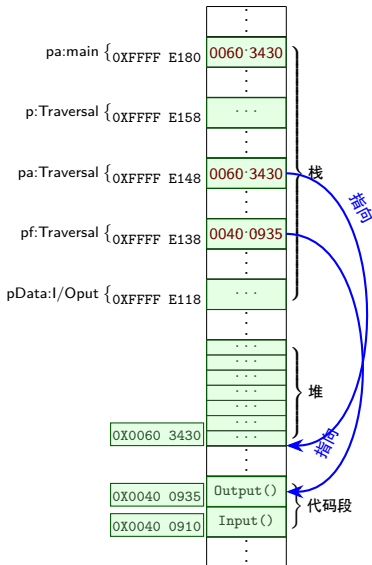
► 动态数组作函数参数

```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++){
        pf(p); //调用传入的函数
    }
}

void Input(int *pData)
{
    scanf("%d", pData);
}

void Output(int *pData)
{
    printf("%d ", *pData);
}
```



► 动态数组作函数参数

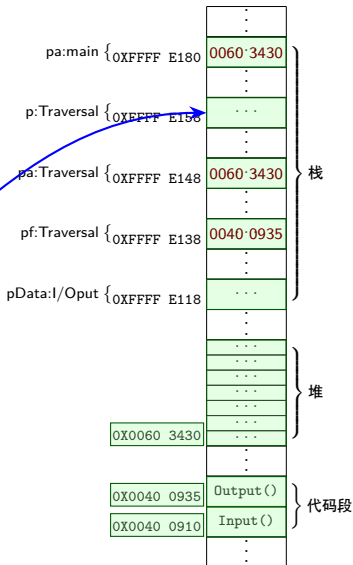
```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++){
        pf(p); //调用传入的函数
    }
}

void Input(int * pData)
{
    scanf("%d", pData);
}

void Output(int * pData)
{
    printf("%d ", *pData);
}
```

声明指针变量



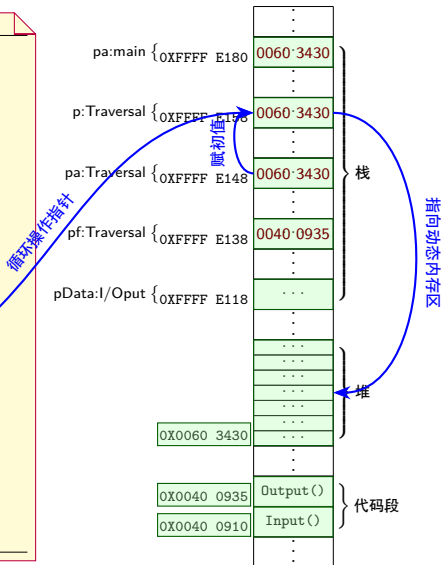
► 动态数组作函数参数

```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++){
        pf(p); //调用传入的函数
    }
}

void Input(int *pData)
{
    scanf("%d", pData);
}

void Output(int *pData)
{
    printf("%d ", *pData);
}
```



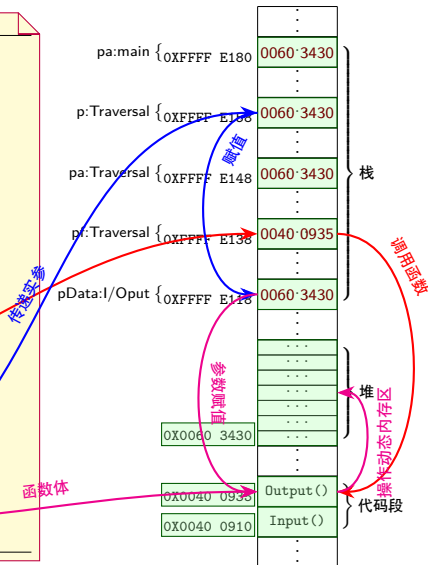
► 动态数组作函数参数

```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++) {
        pf(p); //调用传入的函数
    }
}

void Input(int *pData)
{
    scanf("%d", pData);
}

void Output(int *pData)
{
    printf("%d ", *pData);
}
```



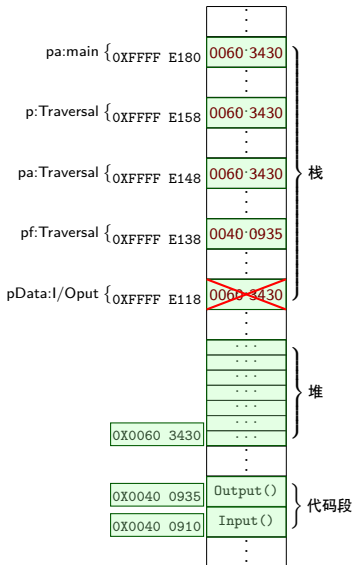
► 动态数组作函数参数

```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++){
        pf(p); //调用传入的函数
    }
}

void Input(int * pData)
{
    scanf("%d", pData);
}

void Output(int * pData)
{
    printf("%d ", *pData);
}
```



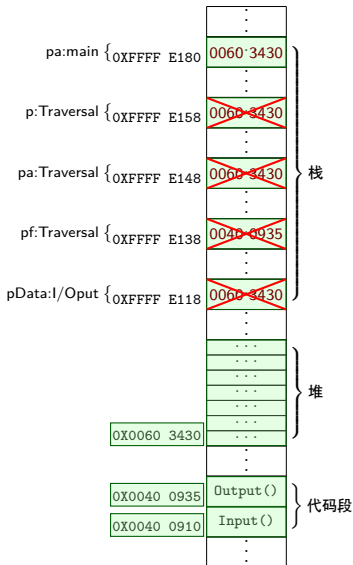
► 动态数组作函数参数

```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int *))
{
    int *p;
    for(p = pa; p < pa + n; p++){
        pf(p); //调用传入的函数
    }
}

void Input(int *pData)
{
    scanf("%d", pData);
}

void Output(int *pData)
{
    printf("%d ", *pData);
}
```



▶ 动态数组作函数参数

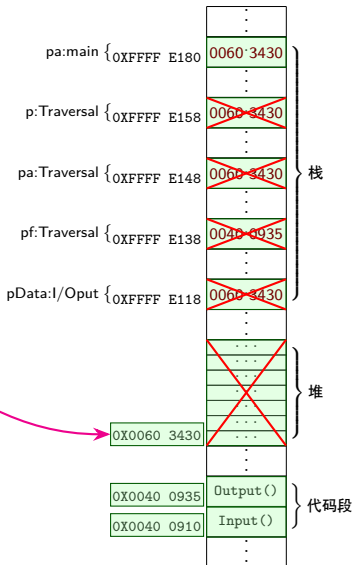
```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; //防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int*))
{
    int *p;
    for(p = pa; p < pa + n; p++){
        pf(p); // 调用传入的函数
    }
}

void Input(int *pData)
{
    scanf("%d", pData);
}

void Output(int *pData)
{
    printf("%d ", *pData);
}
}
```

释放动态内存区



► 动态数组作函数参数

```
#include <stdio.h>
#include <stdlib.h>
...
int main()
{
    int *pa = NULL;
    ...
    /* 分配内存 */
    pa = (int*)malloc(n * sizeof(int));
    ...
    Traversal(pa, n, Input);
    ...
    Traversal(pa, n, Output);
    ...
    // 释放内存
    free(pa);
    pa = NULL; // 防止悬空指针
    return 0;
}

void Traversal(int *pa, int n, void (*pf)(int*))
{
    int *p;
    for(p = pa; p < pa + n; p++){
        pf(p); // 调用传入的函数
    }
}

void Input(int *pData)
{
    scanf("%d", pData);
}

void Output(int *pData)
{
    printf("%d ", *pData);
}
}
```

避免悬空指针

释放动态内存区

