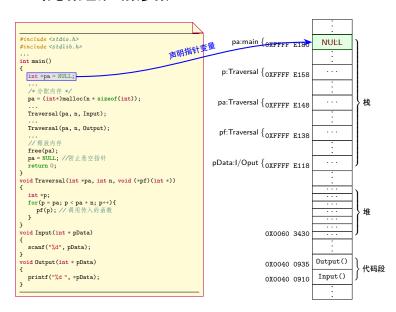
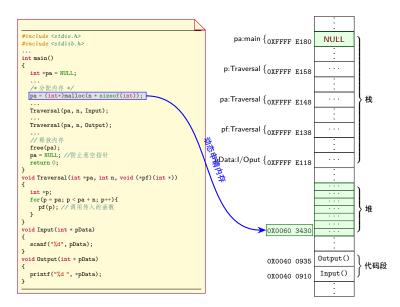
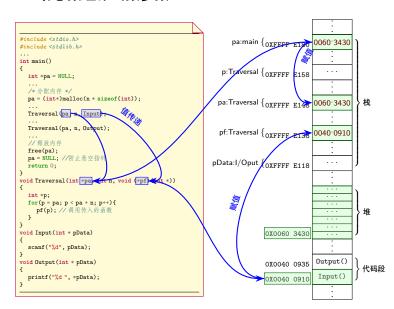
动态数组作函数参数

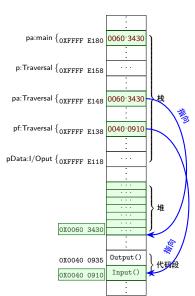


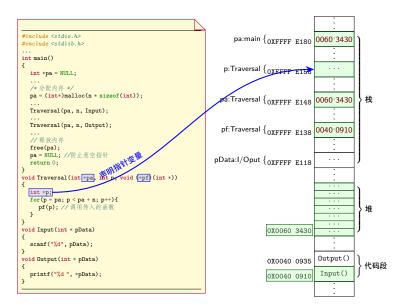


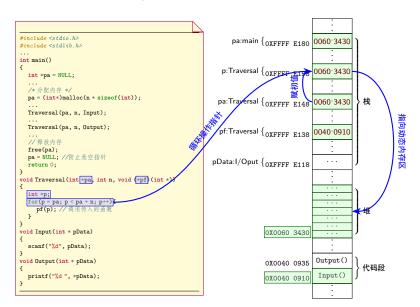
```
#include <etdio h>
                                                            pa:main {0XFFFE E180 0060:3430
#include <stdlib.h>
int main()
                                                         p:Traversal OXFFFF E158
  int *pa = NULL;
  /* 分配内存 */
  pa = (int*)malloc(n * sizeof(int));
                                                        pa:Trayersal {0XFFFF E148
                                                                                                  栈
  Traversal(pa, n, Input);
  Traversal(pa, n, Output);
                                                        #: Traversal { OXFFFF E138
  // 释放内存
  free(pa);
  pa = NULL; //防止悬空指针
                                                      pData:I/Oput {OXFFFF E118
  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
                                                                                                  堆
void Input(int * pData)
                                                                      0X0060 3430
  scanf("%d", pData);
                                                                                    Output()
void Output(int * pData)
                                                                      0X0040 0935
                                                                                                  代码段
  printf("%d", *pData);
                                                                                     Input()
                                                                      0X0040 0910
```

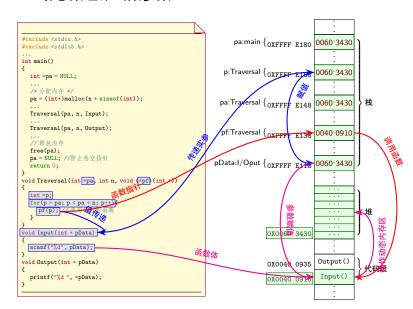


```
#include <stdio h>
#include <stdlib.h>
int main()
  int *pa = NULL;
  /* 分配内存 */
  pa = (int*)malloc(n * sizeof(int));
  Traversal(pa, n, Input);
  Traversal(pa, n, Output);
  // 释放内存
  free(pa);
  pa = NULL; //防止悬空指针
  return 0;
void Traversal(int *pa, int n, void (*pf)(int *))
  int *p:
  for(p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



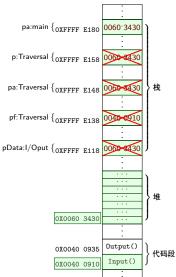


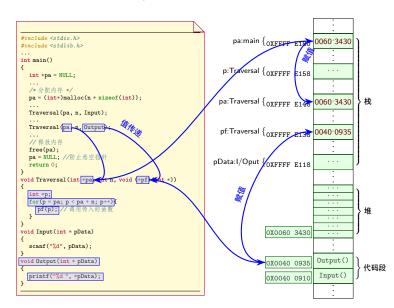




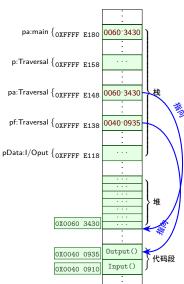
```
#include <stdio h>
                                                             pa:main {0XFFFF E180 0060 3430
#include <stdlib.h>
int main()
                                                          p:Traversal {0XFFFF E158 | 0060:3430
  int *pa = NULL;
  /* 分配内存 */
  pa = (int*)malloc(n * sizeof(int));
                                                         pa:Traversal {0XFFFF E148 | 0060·3430 |
  Traversal(pa, n, Input);
  Traversal(pa, n, Output);
                                                         pf:Traversal {0XFFFF E138 | 0040:0910
  // 释放内存
  free(pa);
  pa = NULL; //防止悬空指针
                                                       pData:I/Oput {OXFFFF E118 0
  return 0;
void Traversal(int *pa, int n, void (*pf)(int *))
  int *p;
  for(p = pa; p < pa + n; p++)
                                                                                                   堆
void Input(int * pData)
                                                                       0X0060 3430
  scanf("%d", pData);
                                                                                      Output()
void Output(int * pData)
                                                                       0X0040 0935
                                                                                                   代码段
                                                                                      Input()
  printf("%d ", *pData);
                                                                       0X0040 0910
```

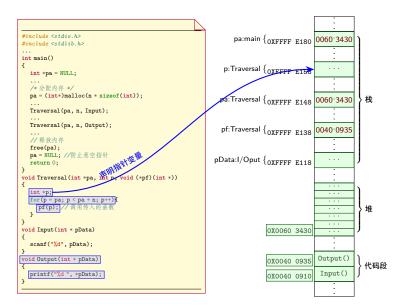
```
#include <etdio h>
#include <stdlib.h>
int main()
  int *pa = NULL;
  /* 分配内存 */
  pa = (int*)malloc(n * sizeof(int));
  Traversal(pa, n, Input);
  Traversal(pa, n, Output);
  // 释放内存
  free(pa);
  pa = NULL; //防止悬空指针
  return 0;
void Traversal(int *pa, int n, void (*pf)(int *))
  int *p;
  for(p = pa; p < pa + n; p++)
void Input(int * pData)
  scanf("%d", pData);
void Output(int * pData)
  printf("%d", *pData);
```

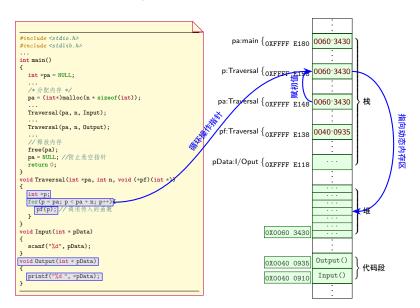


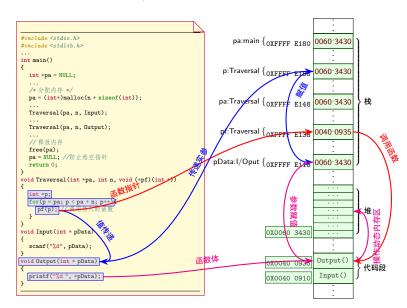


```
#include <stdio h>
#include <stdlib.h>
int main()
  int *pa = NULL;
  /* 分配内存 */
  pa = (int*)malloc(n * sizeof(int));
  Traversal(pa, n, Input);
  Traversal(pa, n, Output);
  // 释放内存
  free(pa);
  pa = NULL; //防止悬空指针
  return 0;
void Traversal(int *pa, int n, void (*pf)(int *))
  int *p;
  for(p = pa; p < pa + n; p++)
void Input(int * pData)
  scanf("%d", pData);
void Output(int * pData)
```

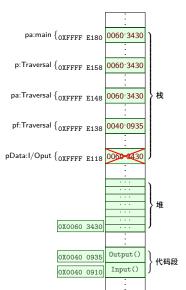








```
#include <stdio h>
#include <stdlib.h>
int main()
  int *pa = NULL;
  /* 分配内存 */
  pa = (int*)malloc(n * sizeof(int));
  Traversal(pa, n, Input);
  Traversal(pa, n, Output);
  // 释放内存
  free(pa);
  pa = NULL; //防止悬空指针
  return 0;
void Traversal(int *pa, int n, void (*pf)(int *))
  int *p;
  for(p = pa; p < pa + n; p++)
void Input(int * pData)
  scanf("%d", pData);
void Output(int * pData)
```



```
#include <etdio 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++)
void Input(int * pData)
  scanf("%d", pData);
void Output(int * pData)
  printf("%d ", *pData);
```

