

```
#include <stdlib.h>
```

```
typedef struct {  
    int n3;  
    double *arr3;  
    double coeff3;  
} mainstruct_lev3_t;
```

```
typedef struct {  
    int n2;  
    mainstruct_lev3_t *arr2;  
    double coeff2;  
} mainstruct_lev2_t;
```

```
typedef struct {  
    int n;  
    mainstruct_lev2_t *arr;  
    double coeff;  
} mainstruct_t;
```

```
#define N 10000
```

```
int main() {  
    mainstruct_t *main_p;  
  
    main_p = (mainstruct_t*) malloc(sizeof(mainstruct_t));  
    main_p->arr = (mainstruct_lev2_t*) malloc(sizeof(mainstruct_lev2_t));  
    main_p->arr->arr2 = (mainstruct_lev3_t*) malloc(sizeof(mainstruct_lev3_t));
```

```
    main_p->arr->arr2->n3 = N;  
    main_p->arr->arr2->arr3 = (double*) malloc(sizeof(double) * main_p->arr->arr2->n3);
```

```
    #pragma acchelper declare(main_p->arr->arr2->n3{int},  
main_p->arr->arr2->arr3{double*:restrictconst})
```

```
    #pragma acchelper region begin  
    #pragma acc parallel loop copy(main_p->arr->arr2->arr3  
[0:main_p->arr->arr2->n3])  
    for(int i=0;i<main_p->arr->arr2->n3;i++) {  
        main_p->arr->arr2->arr3[i] = i;  
    }
```

```
#pragma acchelper region end
```

```
    return main_p->arr->arr2->arr3[0] * main_p->arr->arr2->arr3[N-1] != 0;  
}
```