

allocate, deallocate

```
#include <stdio.h>
#include <stdlib.h>
void* allocate_vector(int size, int m)
```

```
{
    void *a;
    if ( ( a=(void *)malloc( m * size ) ) == NULL ) {
        fprintf(stdout, "Error:Memory does not enough! in vector %n");
        exit(1);
    }
    return a;
}
```

```
void deallocate_vector(void *a)
{
    free( a );
}
```

```
void** allocate_matrix(int size, int m, int n)
{
    void **aa;
    int i;
    if ( ( aa=(void **)malloc( m * sizeof(void*) ) ) == NULL ) {
        fprintf(stdout, "Error:Memory does not enough! aa in matrix %n");
        exit(1);
    }
    if ( ( aa[0]=(void *)malloc( m * n * size ) ) == NULL ) {
        fprintf(stdout, "Error:Memory does not enough! in matrix %n");
        exit(1);
    }
    for(i=1;i<m;i++) aa[i]=(char*)aa[i-1]+size*n;
    return aa;
}
```

```
void deallocate_matrix(void **aa)
{
    free( aa );
}
```

Same interface with FORTRAN

```
allocate_vector vector[:]
allocate_matrix matrix[:,:]
```