

```
#include <stdio.h>
```

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

```
int main (char argc, char * argv[]) {
```

```
    float a[argc-1], i, sum = 0;
```

```
    for (i = 1; i < argc; i++) {
```

```
        a[i-1] = atof(argv[i]);
```

```
        sum += a[i-1];
```

```
    }
```

```
    printf("输出的总和为: %f", sum);
```

```
    return 0
```

```
}
```

```
#include <stdio.h>
```

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

```
void make (int **p, int m, int n) {
```

```
    int i, j;
```

```
    srand (time (NULL));
```

```
    for (i=0; i<m; i++)
```

```
        for (j=0; j<n; j++)
```

```
            p[i][j] = rand() % 100;
```

```
}
```

```
void print (int **p, int m, int n) {
```

```
    int i, j;
```

```
    printf ("生成的二维数组如下\n");
```

```
    for (i=0, i<m, i++) {
```

```
        for (j=0, j<n, j++)
```

```
            printf ("%d\t", &p[i][j]);
```

```
        printf ("\n");
```

```
    }
```

```
}
```

```
int main () {
```

```
    int m, n, **p
```

```
    printf ("请输入二维数组的行数:");
```

```
    scanf ("%d", &m);
```

```
    printf ("请输入二维数组的列数:");
```

```
    scanf ("%d", &n);
```

```
p = (int **) malloc(m * sizeof(int *));
```

```
if (!p) {
```

```
    printf("空间不足");
```

```
    exit(1);
```

```
}
```

```
for (i = 0; i < m; i++) {
```

```
    *(p+i) = (int *) malloc(n * sizeof(int));
```

```
    if (!(p+i)) {
```

```
        printf("空间不足");
```

```
        exit(1);
```

```
    }
```

```
}
```

```
make(p, m, n);
```

```
print(p, m, n);
```

```
for (i = 0; i < m; i++)
```

```
    free(*(p+i))
```

```
free(p)
```

```
return 0
```

```
}
```

```
#include <stdio.h>
```

```
struct node {
```

```
    int num;
```

```
    struct node *next;
```

```
};
```

```
struct node * CreateList(void) {
```

```
    struct node *head = NULL, *p;
```

```
    int num;
```

```
    printf("输入数据,输入0时结束创建:");
```

```
    scanf("%d", &num);
```

```
    while (!num) {
```

```
        p = (struct node *) malloc(sizeof(struct node));
```

```
        p->num = num;
```

```
        p->next = head;
```

```
        head = p;
```

```
        printf("输入数据,输入0时结束创建:");
```

```
        scanf("%d", &num);
```

```
    }
```

```
    return head;
```

```
}
```

```
void print (int *head) {
```

```
    int *p = head;
```

```
    while (p->next != NULL)
```

```
        printf("%d\n", p->num);
```

```
}
```

```
int main ( ){  
    struct node *p  
    p = CreateList();  
    print(p);  
    return 0;  
}
```