

# 三元组（结构体实现）

```
#include <stdio.h>

#include <stdlib.h>

#define ok 1

#define error 0

typedef float Elemtyp;
typedef Elemtyp *Triplet;
void initTriplet (Triplet &T, Elemtyp v[])
{
    int i;
    T=(Elemtyp *)malloc(3*sizeof(Elemtyp));
    for (i=0; i < 3; i++){
        T[i] = v[i];
    }
}
void DestroyTriplet (Triplet &T)
{
    free (T);
    T = NULL;
}
int Get (Triplet T,int i,Elemtyp &a){
    if(i >= 1&& i<= 3){
        a = T[i-1];
        return ok;
    }
    else{
        return error;
    }
}
int Put (Triplet T, int i, Elemtyp a){
    if (i >= 1&&i <= 3){
        T[i-1] = a;
        return ok;
    }
    else {
        return error;
    }
}
int IsAscending (Triplet T){
    if (T[0] <= T[1] && T[1] <= T[2])
        return ok;
}
int IsDescending (Triplet T){
    if (T[0] >= T[1] && T[1] >= T[2])
        return error;
}
void mulCode (Triplet T, int k){
    int i;
```

```

        for (i=0;i < 3;i++){
            T[i] = T[i]*k;
            printf ("%f ", T[i]);
        }
        printf ("\n");
    }
float Max (Triplet T){
    float max;
    if (T[0] >= T[1]){
        max = T[0];
    }
    else {
        max = T[1];
    }
    if (max <= T[2]){
        max = T[2];
    }
    return max;
}
void add (Triplet &T, Triplet T2){
    int i;
    for (i=0;i < 3;i++){
        T[i] += T2[i];
    }
}
int main(){
    Triplet T, T2;
    float v[3], x, e;
    int i, n, k;
    printf ("输入三元组\n");
    for (i=0;i < 3;i++){
        scanf ("%f", &v[i]);
    }
    initTriplet (T, v);
    printf ("三元组中最大值\n");
    printf ("%f\n", Max(T));
    printf ("改变三元组中的值\n");
    scanf ("%d%f", &n, &x);
    Put (T, n, x);
    printf ("得到三元组的值\n");
    scanf ("%d", &n);
    Get (T, n, e);
    printf ("%f\n", e);
    printf ("判断升降序\n");
    if (IsDescending (T)){
        printf ("升序\n");
    }
    else {
        printf ("降序\n");
    }
    printf ("给三元组同乘比例系数\n");
    scanf ("%d", &k);
    mulCode (T, k);
    printf ("定义一个新的三元组\n");
    for (i=0;i < 3;i++){
        scanf ("%f", &v[i]);
    }
    initTriplet (T2, v);

```

```
printf ("两个三元组做和\n");  
add (T, T2);  
for (i=0;i < 3;i++){  
    printf ("%f ", T[i]);  
}  
DestroyTriplet (T);  
return 0;  
}
```