7. Arrays-basic info
能存储**any type**: int, double, <u>char, struct, array</u>…

Assume there is an array A[5]

1. 必须先declare array的大小

2. Cannot assign a number/array to A directly, i.e. A={1,2,3,4,5} / A=4. (如果想要复制一个array 用 for loop). Note: array and for loop is closely associated

3. 5 means the number of elements in A, 但是因为index from 0, 所以index最多可以到4, i,e, A[4]

```
#define N 5

int
main(int argc, char *argv[]) {
    int A[N], i;
    for (i=0; i<N; i++) {
        A[i] = i*(i+1)/2;
    }
    for (i=0; i<N; i++) {
        printf("A[%d]=%d  ", i, A[i]);
    }
    printf("\n");
    return 0;
}
```

```
A[0]=0  A[1]=1  A[2]=3  A[3]=6  A[4]=10
```

4. Error occurs (segmentation fault, 一般都是有关内存的问题的错误提示) when refer to array elements that have not been declared i.e. A[5], A[-1] (对没错在C中不可以用negative index) 或者 refer to higher dimension of the array i.e. A[1][2]. Note: compiler won't have bound check, so it won't tell you which array is referred out of range==需要自己一步步check

5. Example: 读取一组数据到array中，注意超出边界的处理以及scanf的运用

```
    printf("Enter as many as %d values, ^D to end\n",
        maxvals);
    n = 0; excess = 0;
    while (scanf("%d", &next)==1) {
        if (n==maxvals) {
            excess = excess+1;
        } else {
            A[n] = next;
            n += 1;
        }
    }
    printf("%d values read into array", n);
    if (excess) {
        printf(", %d excess values discarded", excess);
    }
    printf("\n");
```

```
mac: ./readarray
Enter as many as 10 values, ^D to end
1 2 3 4 3
^D
5 values read into array
mac: ./readarray
Enter as many as 10 values, ^D to end
1 2 3 4 3 4 5 4 3 4 5 6 7
^D
10 values read into array, 3 excess values discarded
```

6. Insertion Sort: bad performance in general but excellent when input array is already sorted

核心思想：application of decrease-by-one technique, we assume that the smaller subarray has been already sorted, then we just need to find the next element in the original array to put it in the right position

第一种思路（书上必会）：将当前unsorted array中的最小元素放到sorted subarray中的最后

```
/* assume that A[0] to A[n-1] have valid values */
for (i=1; i<n; i++) {
    /* swap A[i] left into correct position */
    for (j=i-1; j>=0 && A[j+1]<A[j]; j--) {
        /* not there yet */
        int_swap(&A[j], &A[j+1]);
    }
}
/* and that's all there is to it! */
```

```
mac: ./insertionsort
Enter as many as 10 values, ^D to end
1 8 15 3 17 12 4 8 4
^D
9 values read into array
Before:    1   8   15   3   17   12   4   8   4
After :    1   3    4   4    8    8   12  15  17
```

第二种思路（升级版，减少int swap的次数）：
```
for ( i=1; i<n; i++) {
    value = A[i];
    j = i - 1;
    while j>=0 and A[j] > value {
        A[j+1] = A[j];
        j -= 1;
    }
    A[j+1] = value;
}
```

Analysis:

Best Case: The array is almost sorted: only need to do a few swaps => $O(n)$
Worst Case: The array is reversely sorted: each element need to be swapped in linear time => $O(n^2)$

7. A itself represents a pointer towards the first element of A, which is equivalent to &A[0]—i.e. 如果我们需要一个pointer, p指向A中的第一个元素：p=A (等价于 p=&A[0])
    同样的，如果需要指向第n个元素：p=A+n-1 (等价于 p=&A[n-1])

```
int A[N], *p, i;
for (i=0; i<N; i++) {
    A[i] = i*(i+1)/2;
}
printf("A = %10p\n", A);
p = A;
for (p=A; p<A+N; p++) {
    printf("p = %10p, *p = %2d\n", p, *p);
}
```

```
A = 0xbffff5e4
p = 0xbffff5e4, *p =  0
p = 0xbffff5e8, *p =  1
p = 0xbffff5ec, *p =  3
p = 0xbffff5f0, *p =  6
p = 0xbffff5f4, *p = 10
```

8. Array argument can be declared as either arrays i.e. A[5], or as pointers of the same underlying type i.e int* p. 这里可以这么想：Array其实是将一堆同样type的Variable存放在一起，每个element占据的空间是一样的，所以其实只需要知道第一个element的位置就可以 Also, the number of elements is not need to be part of declaration i.e. A[ ]

```
typedef double vector_t[SIZE];

void
vector_add(vector_t A, vector_t B, vector_t C, int n) {
    int i;
    for (i=0; i<n; i++) {
        C[i] = A[i] + B[i];
    }
}
```

9. 当一个array作为argument传入一个方程，改变其中的元素可以永久改变这个array中的元素 (have global effects). 但如果只是改变Array 指针的指向，并不能永久改变Array的指向（local effects）

10. typedef --自己define一种type, 简化后面的declaration, 并且大大滴方便如果需要修改这种type 也可以直接 typedef double data_t, 如果想要改变一些data_t 的Variable type, 直接在typedef里修改

Note: a convention， 如果typedef一种变量， 一般在type名后加_t， i.e. data_t (不加_t 扣分🤟)

11. 2D Array

    e.g. int (Y[5])[10]

    只有dominant dimension can be left unspecified--5 因为可以通过subdominant dimension 推算出现在的dominant dimension

应用：矩阵相加

```
typedef vector_t sqmatrix_t[SIZE];

void
sqmatrix_add1(sqmatrix_t A, sqmatrix_t B, sqmatrix_t C,
        int n) {
    int i,j;
    for (i=0; i<n; i++) {
        for (j=0; j<n; j++) {
            C[i][j] = A[i][j] + B[i][j];
        }
    }
}

void
sqmatrix_add2(vector_t A[], vector_t B[], vector_t C[],
        int n) {
    int i;
    for (i=0; i<n; i++) {
        vector_add(A[i], B[i], C[i], n);
    }
}
```