

Массивы в языке C

Практическое занятие

Список инициализаторов

- $T \text{ arr}[4] = \{0,1,2,3\}$
 - $T \text{ arr}[7] = \{?$
- $T \text{ arr}[N=2][M=5] = \{\{0,1,2,3,4\},\{0,1,2,3,4\}\}$
 - $T \text{ arr}[N=4][M=6] = \{?$
- $T \text{ arr}[N=3][M=4][L=2] = \{$
 - $\{\{0,1\},\{0,1\},\{0,1\},\{0,1\}\},$
 - $\{\{2,3\},\{2,3\},\{2,3\},\{2,3\}\},$
 - $\{\{4,5\},\{4,5\},\{4,5\},\{4,5\}\}$
 - $\}$
 - $T \text{ arr}[N=2][M=3][L=4] = \{?$

Многомерные массивы

- Вычислите адрес 5-го эл-та 1-о мерного массива **ar1[N]**:
$$\text{адр}[i] = \text{адр}[0] + i * \text{sizeof}(T);$$
- Вычислите адрес (4,7) –го эл-та 2-мерного массива **ar2[N][M]** , **M = 8**
$$\text{адр}[i][j] = \text{адр}[0][0] + i * M * \text{sizeof}(T) + j * \text{sizeof}(T);$$
- Вычислите адрес (2,3,5) –го эл-та 3-мерного массива **ar3[N][M][K]**, M=5, K=6
$$\text{адр}[i][j][k] = \text{адр}[0][0][0] + i * M * K * \text{sizeof}(T) + j * K * \text{sizeof}(T) + k * \text{sizeof}(T);$$

К i-му элементу массива arr[N]
можно обратиться как по имени,
так и по указателю:

```
int tmp = arr[i];  
int tmp = p[i];  
int tmp = *(p+i);  
int tmp = *(arr+i);
```

К i, j-му элементу массива
ar2[N][M] можно обратиться
как по имени, так и по указателю:

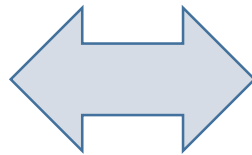
```
int tmp = ar2[i][j];  
int tmp = p[i][j];  
int tmp = *(p+i)[j];  
int tmp = *(ar2+i)[j];  
int tmp = *(* (ar2+i)+j);  
int tmp = *(ar2[i] + j);  
... дополните .
```

К i, j, k-му элементу массива
ar3[N][M][K]
можно обратиться как по имени,
так и по указателю:

```
int tmp = ar3[i][j][k];  
...  
tmp = p[i][j][k]  
Дополните...
```

Сортировка методом выбора (указатели)

```
int ar[] = {4,1,-10, 55, 2, -5};
size_t n = sizeof(ar)/sizeof(ar[0]);
int * pCur = ar;
for (int i = 0; i < (n-1); ++i){
    int * pMin = pCur;
    int * pTmp = pCur + 1;
    for (int j = i + 1; j < n; ++j){
        if(*pTmp < *pMin)
            pMin = pTmp;
        pTmp++;
    }
    int tmp = *pMin;
    *pMin = *pCur;
    *pCur = tmp;
    pCur++;
}
```



```
for(int i = 0; i < n-1; ++i){
    int min = i;
    for(int j=i+1; j < n; ++j){
        if(ar[j] < ar[min]) min = j;
    }
    int temp = ar[min];
    ar[min] = ar[i];
    ar[i] = temp;
}
```

Сортировка методом пузырька.

Просматриваем массив снизу вверх и меняем стоящие рядом элементы если «Нижний» меньше чем «Верхний». Самый маленький элемент окажется вверху. Повторим все для $n-1$ оставшихся элементов, лежащих ниже самого маленького.

исходный массив		обмен 2 и 3		обмен 2 и 7		обмен 2 и 5		нет обмена
1		1		1		1		1
5		5		5		2		2
7		7		2		5		5
3		2		7		7		7
2		3		3		3		3
первый проход циклом по массиву								

```
int ar[]={6,1,-10, 5, 2, -7};
```

```
for( int i = 0; i < size; i ++ ) { // i - номер прохода
    for( j = size - 1; j > i; j-- ) {
        if ( a[j-1] > a[j] ) {
            x = a[j-1];
            a[j-1] = a[j];
            a[j] = x;
        }
    }
}
```

Исправить код

Дана строка символов. Необходимо убрать из нее все пробелы за 1 проход.

```
int main(){
    char * pStr = "asdfn ddf ddd ffff";
    printf("%s\n", pStr);
    char* str = pStr;
    while (*str != '\0'){
        if (*str != ' '){
            *pStr = *str;
            *pStr++;
        }
        str++;
    }
    *pStr = '\0';
    printf("%p\n", pStr);
}
```

Домашняя работа.

1. Создать массив [5][7]. Инициализируем числами от 0 до 35:
 - a) при помощи списка инициализаторов;
 - b) в цикле ;
 - c) с использованием вложенного цикла.
2. Сортировка методом пузырька при помощи указателей.
3. Изучить вопрос создания многомерного динамического массива.
4. Задан массив из 30 вещественных чисел. Найти элемент (индекс), имеющий максимальное значение в массиве.

* Вычисление значения арифметического выражения, введенного с клавиатуры. Использовать операции: +, -