Технологии программирования

Введение.

Массивы, матрицы, строки

Контакты

- Смирнов Иван Валентинович, доцент кафедры информационных технологий
- smirnov_iv@pfur.ru

Технологии программирования

- 3 темы (C++):
  - Структуры (массивы, списки) – 3 занятия
  - Основы ООП (классы) – 3 занятия
  - Наследование и полиморфизм – 3 занятия
- На каждом занятии решаем задачу
  - Необходима среда DEVC++

- По каждой теме индивидуальное домашнее задание
  - Результат – код на C++, компилирующийся в DEVC++
  - Вопросы по любой строчке
  - Критерии – работающий код и ответы на все вопросы

- 10 баллов – посещение
- 15 баллов – активность на занятиях
- 25 баллов за каждое домашнее задание

Массивы

- Массив– это группа ячеек памяти одинакового типа, расположенных рядом и имеющих общее имя. Каждая ячейка в группе имеет уникальный номер.

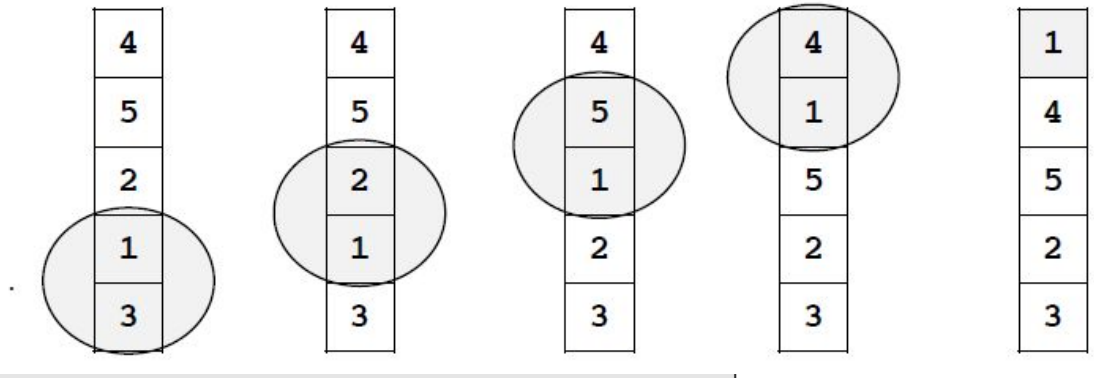
```
int A[10], B[20];   // 2 массива на 10 и 20 целых чисел
float C[12];        // массив из 12 вещественных чисел
```

Элементы массивов в языке Си нумеруются с нуля. Таким образом, если в массиве 10 элементов, он содержит элементы:

```
A[0], A[1], A[2], ..., A[9]
```

```c
#include <stdio.h>
const int N = 10;       // размер массива
main()
{
int i, A[N];            // объявление массива
printf("Введите массив A\n");  // подсказка для ввода
for ( i = 0; i < N; i ++ ) {        // цикл по всем элементам
   printf("Введите A[%d]> ", i );  // подсказка для ввода A[i]
   scanf ("%d", &A[i]);            // ввод A[i]
   }
for ( i = 0; i < N; i ++ ) // цикл по всем элементам
   A[i] = A[i] * 2;        // умножить A[i] на 2
printf("\nРезультат:\n");
for ( i = 0; i < N; i ++ ) // цикл по всем элементам
   printf("%d ", A[i]);    // вывести A[i]
}
```

Сортировка




```
#include <stdio.h>
const int N = 10;
main()
{
int i, j, A[N], c;

   // здесь надо ввести массив A
for ( i = 0; i < N-1; i ++ ) // достаточно поставить N-1 элементов
  for ( j = N-2; j >= i; j -- ) // идем с конца массива в начало
      if ( A[j] > A[j+1] )     // если они стоят неправильно, ...
        {
        c = A[j]; A[j] = A[j+1]; // переставить A[j] и A[j+1]
        A[j+1] = c;
        }
printf("\n Отсортированный массив:\n");
for ( i = 0; i < N; i ++ )
   printf("%d ", A[i]);
}
```

Поиск элемента

```
#include <stdio.h>
const int N = 10;
main()
{
int L = 0, R = N-1, m, A[N],
    flag = 0;   // переменная-флаг, нашли (1) или нет (0)

   // здесь надо ввести массив A
   // и отсортировать его по возрастанию

printf("Введите искомый элемент\n");
scanf( "%d", &x );
```

```
while ( L <= R ) {
   m = (L + R) / 2;     // середина интервала
   if ( A[m] == x ) {   // нашли нужный элемент
      flag = 1;         // установить флаг
      break;            // выйти из цикла
      }
   if ( x < A[m] ) R = m - 1; // сужаем границы области поиска
   else            L = m + 1;
   }
```

```
if ( flag )
     printf ( "Нашли: A[%d]=%d", m, A[m] );
else printf ( "Такого элемента нет" );
}
```

Строки

Символьная строка – это последовательность символом, которая заканчивается **символом с кодом 0**.

Символ с кодом ноль (обозначается как **'\0'**) и цифра ноль (обозначается **'0'**, имеет код 48) – это два разных символа.

```
#include <stdio.h>
main()
{
   char Name[50];
   printf("Как тебя зовут? ");
   scanf("%s", Name);
   printf("Привет, %s!", Name);
}
```

Заменить А на Б

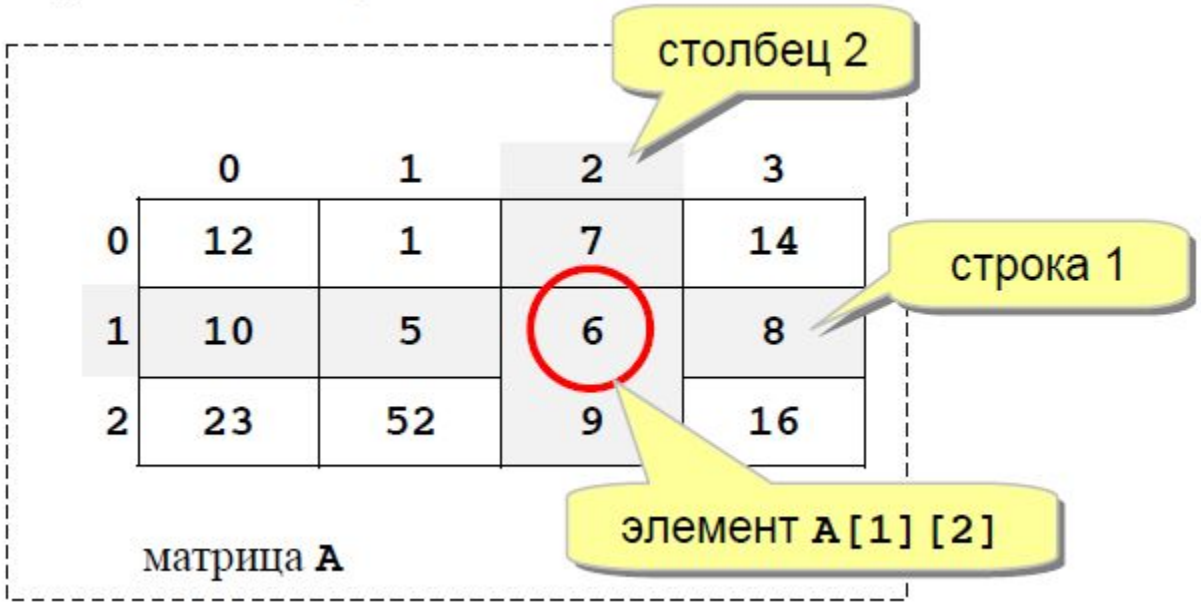
```
#include <stdio.h>
main()
{
  char s[80];
  int i;

  printf( "\n Введите строку \n" );

  gets ( s );
  i = 0; // начать с первого символа, s[0]
  while ( s[i] != '\0' ) // пока не достигли конца строки
    {
    if ( s[i] == 'А' )    // если очередной символ - 'А', ...
      s[i] = 'Б';         // меняем его на 'Б'
    i ++;                 // переходим к следующему символу
    }
  puts ( "Результат:\n" );
  puts ( s );
}
```

Функции string.h

- Длина строки – strlen(s)
- Сравнение строк – strcmp(s1, s2)
- Копирование строк – strcpy(s2, s1) // s2 (приемник) <- s1 (источник)
- Объединение строк - strcat(s1, s2) // дописать s2 в конец s1

Матрицы

- Матрица – многомерный массив




```
int Bmv[20][10];
```

```
float A[M][N], i, j, row, col;
...
row = col = 0;   // сначала считаем, что A[0][0] - минимальный
for ( i = 0; i < M; i ++ )      // просмотр всех строк
   for ( j = 0; j < N; j ++ )  // просмотр всех столбцов
      if ( A[i][j] < A[row][col] ) {
           row = i;             // запомнили новые индексы
           col = j;
           }
 printf ("Минимальный элемент A[%d][%d]=%d",
                        row, col, A[row][col]);
```

Массивы строк

```
char s[4][80] = { "Вася", "Петя" };
```

```
char s[20][80];
int i, count = 0;

printf("Введите текст (Enter - закончить)\n");

for ( i = 0; i < 20; i ++ ) {    // пытаемся прочитать 20 строк
   gets (s[count]);              // прочитать строку с клавиатуры
   if (s[count][0] == '\0')      // если введена пустая строка,
      break;                     // то выйти из цикла
   count ++;
   }
```

```
void SortStrings ( char *s[], int n ) // *s[] - массив указателей
{                                     // n - число строк
   char *p;
   int i, j;
   for ( i = 0; i < n-1; i ++ )
      for ( j = n-1; j > i; j -- )
         if ( strcmp(s[j-1],s[j]) > 0 ) {
           p = s[j]; s[j] = s[j-1];  // перестановка УКАЗАТЕЛЕЙ
           s[j-1] = p;
           }
}
```

Задача

Написать программу для выполнения следующих действий:

1. Ввести массив строк с экрана (размер не более 20)
2. Отсортировать строки по убыванию их длины
3. Ввести длину искомой строки
4. Найти и вывести строки заданной длины