

**Лабораторная работа №6**  
по дисциплине «Программирование»,  
2 семестр

**Задание:**

Решить задачу своего варианта из нижеприведенного перечня, создав многомодульное консольное приложение для *MS Windows* на языке *C* или *C++*. При этом:

- 1) Ввод исходных данных выполнить из нестандартного текстового файла.
- 2) Вывод исходных данных и результатов также выполнить в нестандартный текстовый файл.
- 3) Использовать только динамические массивы, не статические. В условиях задач  $N$ ,  $M$ ,  $K$  – переменные, а не константы, и их значения надо также узнать из файла.
- 4) Аномальные ситуации можно не обрабатывать.
- 5) Для решения задачи выделить не менее двух подзадач, не считая подзадач ввода, вывода, выделения/освобождения памяти и проверки аномалий. Все подзадачи оформить в виде функций.
- 6) Внутри функций не использовать глобальных переменных напрямую: вся связь с вызывающей программой только через интерфейс функции – её параметры возвращаемое значение.
- 7) Не смешивать внутри функций решение задачи с вводом или выводом: либо только ввод и/или вывод, либо только решение (поиск, вычисления и т.д.) без ввода дополнительных значений и без вывода результата.
- 8) Все функции расположить не в коде головного модуля программы (рядом с *main*), а в отдельных файлах (\*.h и \*.c/\*.cpp), в одном или нескольких.
- 9) Создать краткий отчет по данной работе в рукописном или печатном виде, из условий задачи и выделенных подзадач и функциональных тестов.

**Пример работы с текстовым файлом и информация** про параметр-массив. функции, прототипы и модули в конце этого файла.

**Перечень задач:**

1. Дана матрица  $C$  из  $N$  ( $N > 1$ ) строк и  $N$  столбцов. Если первая строка не содержит ни одного элемента, значение которого совпадает со значением какого-либо элемента последней строки этой же матрицы, задать значения элементам  $X_1, X_2, \dots, X_N$  по правилу  $X_i = \max(C_{1,i}, C_{N,i})$ ,
2. Дана матрица  $A$  из  $N$  ( $N > 1$ ) строк и  $N$  столбцов. Если в матрице  $A$  элемент с минимальным значением (среди всех элементов этой матрицы) лежит ниже главной диагонали, создать массив  $C_1, C_2, \dots, C_N$  из значений сумм элементов матрицы  $A$ , лежащих в каждом из столбцов с 1 по  $N$  по отдельности.
3. Даны три последовательности:  $X_1, X_2, \dots, X_K$ ;  $Y_1, Y_2, \dots, Y_K$  и  $M_1, M_2, \dots, M_K$ . Каждая тройка элементов  $(X_i, Y_i, M_i)$  представляет параметры одной из  $K$  материальных точек, лежащих в плоскости  $XOY$ : абсциссу  $X_i$ , ординату  $Y_i$  и массу  $M_i$ . Если абсциссы и ординаты всех точек положительны, найти  $(XC, YC)$  – координаты центра тяжести данной системы масс по формулам:

$$XC = \sum_{i=1}^K X_i M_i / \sum_{i=1}^K M_i; YC = \sum_{i=1}^K Y_i M_i / \sum_{i=1}^K M_i$$

4. Дана матрица  $P$  с двумя строками и  $K$  столбцами, каждым столбцом которой задана абсцисса и ордината одной из  $K$  точек плоскости. Если нет ни одной пары точек, расстояние между которыми меньше заданной величины  $R$ , заменить на нуль в матрице  $P$  все отрицательные абсциссы точек, увеличив ординаты этих точек на  $R$ .
5. Дана матрица  $A$  из  $N$  строк и  $M$  столбцов. Если среднее арифметическое матрицы  $A$  положительно, задать элементам  $C_1, C_2, \dots, C_i$  значения тех элементов матрицы  $A$ , которые больше этого среднего арифметического.
6. Дана матрица  $B$  из  $N$  ( $N > 1$ ) строк и  $N$  столбцов. Если в матрице  $B$  элемент с максимальным значением (среди всех элементов этой матрицы) лежит выше главной диагонали, найти сумму элементов матрицы, лежащих ниже главной диагонали.
7. Дана матрица  $A$  из  $N$  строк и  $M$  столбцов. Найти среднее арифметическое элементов матрицы  $A$ , и, если матрица  $A$  не содержит ни одного отрицательного элемента, изменить элементы матрицы путем вычитания из них этого среднего арифметического.
8. Дана матрица  $B$  из  $M$  строк и  $M$  столбцов. Если все элементы главной диагонали матрицы  $B$  отрицательны, разделить все элементы матрицы на максимальный по абсолютной величине элемент матрицы.
9. Дана матрица  $A$  из  $N$  строк и  $N$  столбцов. Если разность максимального и минимального элемента матрицы  $A$  превышает заданную величину  $E$ , заменить в матрице  $A$  все отрицательные элементы нулями, а положительные единицами и подсчитать число выполненных замен.
10. Дана матрица  $B$  из  $N$  строк и  $N$  столбцов и массив  $C_1, C_2, \dots, C_M$ . Если среднее арифметическое элементов  $C_1, C_2, \dots, C_M$  больше минимального элемента матрицы  $B$ , уменьшить на величину последнего каждый из элементов  $C_1, C_2, \dots, C_M$ .
11. Дана матрица  $A$  из  $N$  строк ( $N > 2$ ) и  $M$  столбцов. Если сумма двух первых строк матрицы  $A$  меньше суммы элементов двух последних ее строк, изменить матрицу  $A$ , прибавив к элементам каждой строки заданные элементы  $X_1, X_2, \dots, X_M$ .
12. Дана матрица  $B$  из  $N$  строк ( $N > 1$ ) и  $N$  столбцов. Если ни один из столбцов матрицы  $B$ , не содержит два и более равных нулю элемента, найти сумму элементов матрицы, лежащих на главной диагонали и выше нее.
13. Дана матрица  $A$  из  $N$  строк и  $N$  столбцов. Если ниже главной диагонали матрицы  $A$  нет ни одного отрицательного элемента, изменить матрицу  $A$ , умножив каждый ее элемент на находящийся с ним в одной строке элемент главной диагонали.
14. Дана матрица  $B$  из  $N$  строк и  $M$  столбцов. Если число отрицательных элементов матрицы  $B$  превышает число положительных, увеличить каждый элемент матрицы  $B$  на величину среднего арифметического всех ее элементов.
15. Дана матрица  $A$  из  $N$  строк и  $M$  столбцов. Если сумма элементов последнего столбца матрицы  $A$  положительна, присвоить каждому из элементов  $X_1, X_2, \dots, X_N$  значение среднего арифметического соответствующей по номеру строки матрицы.
16. Дана матрица  $B$  из  $M$  строк и  $M$  столбцов. Кроме матрицы  $B$ , даны элементы  $C_1, C_2, \dots, C_M$ . Если для всех  $i$  и  $j$  выполняется неравенство  $C_i > B_{ij}$ , заменить значение каждого элемента  $C_i$  значением минимального элемента  $i$ -ой строки матрицы  $B$ .
17. Дана матрица  $A$  из  $N$  строк и  $N$  столбцов. Если в матрице  $A$  элементы, равные нулю, встречаются не более, чем в двух строках, задать элементам  $X_1, X_2, \dots, X_N$  значения соответствующих по номеру элементов главной диагонали.
18. Дана матрица  $A$  из  $N$  строк и  $N$  столбцов. Кроме матрицы  $A$ , даны элементы  $C_1, C_2, \dots, C_N$ . Если значения всех этих элементов заключены между заданными значениями  $P$  и  $T$ , получить значения элементов  $X_1, X_2, \dots, X_N$  по формуле 
$$X_i = \sum_{k=1}^N A_{ik}$$

19. Дана матрица  $A$  из  $N$  строк и  $N$  столбцов и массив  $C_1, C_2, \dots, C_N$ . Если среднее арифметическое  $CA$  элементов главной диагонали матрицы  $A$  меньше каждого из элементов  $C_1, C_2, \dots, C_N$ , изменить матрицу  $A$  увеличением положительных ее элементов на величину  $CA$  и уменьшением отрицательных элементов на эту же величину.
20. Дана матрица  $B$  из  $N$  строк и  $N$  столбцов и массив  $C_1, C_2, \dots, C_M$ . Если сумма  $Q$  положительных элементов матрицы  $B$ , превышает абсолютную величину суммы отрицательных из элементов  $C_1, C_2, \dots, C_M$ , увеличить на  $Q$  значение каждого из элементов  $B$  и  $C$ .
21. Дана матрица  $A$  из  $N$  строк и  $N$  столбцов и массив  $C_1, C_2, \dots, C_N$ . Если в последовательности  $C_1, C_2, \dots, C_N$  имеются равные элементы, изменить значения всех ее элементов по правилу:  $C_i = C_i + A_{ii}$ .
22. Дана матрица  $A$  из  $N$  строк и  $N$  столбцов. Кроме матрицы  $A$  дана матрица  $B$  такого же размера. Если каждый элемент матрицы  $A$  больше соответствующего элемента матрицы  $B$ , присвоить элементам  $C_1, C_2, \dots, C_N$  значения по правилу  $C_i = \sum_{k=1}^N (A_{ik} + B_{ik})$
23. Дана матрица  $B$  из  $N$  строк и  $M$  столбцов. Если среднее арифметическое каждого столбца матрицы  $B$  меньше заданной величины  $T$ , заменить значение каждого элемента матрицы  $B$  квадратом этого значения.
24. Дана матрица  $A$  из  $N$  строк и  $N$  столбцов. Кроме матрицы  $A$  даны элементы последовательности  $B_1, B_2, \dots, B_N$ . Если для каждой строки матрицы  $A$  сумма ее элементов ( $P_i$ ) меньше соответствующего элемента последовательности ( $B_i$ ), присвоить всем элементам последовательности значения по правилу:  $B_i = P_i$ .
25. Дана матрица  $A$  из  $N$  строк и  $N$  столбцов. Если разность максимального и минимального элементов каждой строки матрицы  $A$  не превышает заданной величины  $R$ , присвоить каждому из элементов  $C_1, C_2, \dots, C_N$  значение соответствующего по номеру элемента главной диагонали матрицы  $A$ .
26. Дана матрица  $B$  из  $N$  строк и  $N$  столбцов. Если в матрице  $B$  элемент с максимальным значением (среди всех элементов этой матрицы) лежит на главной диагонали, присвоить начальным элементам последовательности  $C_1, C_2, \dots, C_{N^2}$  значения элементов матрицы, лежащих выше главной диагонали, а остальным элементам этой последовательности – значения прочих элементов матрицы.
27. Дана матрица  $A$  из  $N$  строк и  $N$  столбцов. Если в матрице  $A$  нет элементов, абсолютная величина которых отличается от заданной величины  $P$  менее, чем на заданную величину  $E$ , найти для каждой ее строки среднее арифметическое положительных элементов.
28. Дана матрица  $A$  из  $N$  строк и  $N$  столбцов. Кроме матрицы  $A$ , дана матрица  $B$  такого же размера. Если для всех  $i, j$  выполняется неравенство  $A_{ij} + B_{ij} > 0$ , заменить значение каждого элемента матрицы  $A$ , который меньше соответствующего элемента матрицы  $B$ , значением этого элемента матрицы  $B$ .
29. Дана матрица  $B$  из  $N$  строк и  $N$  столбцов. Если  $C_N > \dots > C_3 > C_2 > C_1$ , где  $C_i$ —сумма элементов  $i$ -ой строки матрицы  $B$ , задать элемента  $i$ -ой строки матрицы значения соответствующих элементов  $(i+1)$ -ой строки, а элементам последней строки задать значения элементов первой строки.
30. Даны две последовательности:  $C_1, C_2, \dots, C_M; P_1, P_2, \dots, P_M$ . Если каждый элемент первой последовательности меньше суммы элементов второй, найти при каких значениях  $i, j$  максимально значение выражения  $C_i / (1 + P_j^2 + C_i^2)$
31. Дана матрица  $B$  из  $N$  строк и  $M$  столбцов. Если среди элементов матрицы нет элементов, по абсолютной величине меньших единицы, то изменить матрицу  $B$  путем умножения всех ее элементов на значение последнего элемента матрицы.
32. Дана матрица  $A$  из  $N$  строк и  $N$  столбцов. Если вне главной диагонали не все элементы нулевые, найти такие значения  $i$  и  $j$ , при которых произведение  $A_{ij} \times A_{ji}$  принимает максимальное значение.

33. Дана матрица  $B$  из  $N$  строк и  $N$  столбцов. Если  $C_1 > C_2 > C_3 > \dots > C_N$ , где  $C_j$  – количество неотрицательных элементов в  $j$ -ом столбце матрицы  $B$ , переставить в каждой строке матрицы  $B$  на главную диагональ максимальный из элементов этой строки, увеличив его вдвое.

34. Дана матрица  $A$  из  $N$  строк и  $N$  столбцов. Если в каждой строке матрицы на главной диагонали находится элемент с минимальным по абсолютной величине значением, изменить все элементы матрицы путем вычитания из них значения диагонального элемента из той же строки, где находится изменяемый элемент.

35. Дана матрица  $B$  из  $N$  строк и  $M$  столбцов. Если все элементы первого и последнего столбцов матрицы  $B$  отрицательны, присвоить каждому из элементов  $X_1, X_2, \dots, X_M$  значение суммы отрицательных элементов соответствующего по номеру столбца матрицы.

### Порядок работы с текстовыми файлами:

Способ	Создание нового потока ввода/вывода и определения указателя файла. Позволяет использовать несколько потоков ввода одновременно и несколько потоков вывода.	Переопределение потоков стандартного ввода ( <i>stdin</i> ) и вывода ( <i>stdout</i> ), которые по умолчанию связаны с клавиатурой и экраном для консольного ввода и вывода
Требуется подключить модуль	#include <stdio.h>	#include <stdio.h> #include <conio.h>
Описание переменной	FILE *f;	FILE *f;
Открытие для чтения (r/rt) * **	f = fopen("File1.txt", "r");	f = freopen("File1.txt", "r", stdin)
Открытие для записи с созданием	f = fopen("d:\\File1.txt", "w");	f = freopen("F1.txt", "w", stdout)
Открытие для дозаписи в конец	f = fopen("d:\\Dir\\F1.txt", "a");	f = freopen("File1.txt", "a", stdout)
Чтение чисел (%i,%d,%u, %f,...), символов (%c) и токенов(%s)	fscanf(f, "%d %f", &i, &r);	scanf("%d %f", &i, &r);
Запись чисел	fprintf(f, "i=%d r=%f", i, r);	printf("i=%d r=%f", i, r);
Чтение символов char ch	ch = getc(f);	ch = getch(); ch = getche(); ch=getchar(); или scanf("%c",&ch);
Запись символов	putc(ch, f);	putchar(ch); или printf("%c",ch);
Чтение строк char str[80]	fgets(str, 79, f);	gets(str, 79);
Запись строк	fputs(str, f);	puts(str); или printf("%s\n",str);
Закрытие файла	fclose(f)	fclose(f)
Дополнительные функции	if (feof(f)!=0) конец файла	

\*Относительный путь к файлу указывается от расположения ехе-файла. Например, в папке Debug, Release или папке проекта в зависимости от версии среды программирования и режима компиляции.

\*\*Слеш (\) удваивается, так как является служебным символом в строке, например, \n \t \0 \\\

### Пример 1

Пусть есть файл из 2 строк:

10 11 12

13.1415

Считаем первое, третье и четвертое числа.

```
//-----
#include <stdio.h> // FILE fopen fclose fprintf fscanf, printf, scanf
#include <conio.h> // getch
#include <windows.h> // SetConsoleOutputCP
//-----
void main()
```

```

{
    SetConsoleOutputCP(1251);
    printf("Use \"Lucida Console\\n"); // не забудьте в свойствах консольного окна указать шрифт

    char fn[80];
    printf("Имя файла =?"); // приглашение
    gets(fn); //Ввод строки - имени текстового файла

    FILE *f;
    f = fopen( fn, "rt");
    if (!f) { // или if (f ==NULL) {
        printf("Error: Ошибка при открытии файла %s для чтения\nPress any key", fn);
        getch(); // ожидание нажатия клавиши
        return; // выход из функции main
    }

    int n, *m= new int; // целое и указатель на целое+выделение памяти (с++)
    float z;
    fscanf(f, "%d %d %d", &n, m); // ввод десятичного(%d) n и пропуск следующего, чтение *m
    fscanf(f, "%f", &z); // ввод десятичного(%d) z
    printf("n=%d m=%d z=%7.4f\n", n, *m, z);
// или printf("n=%d m=%d z=%.*f\n", n, *m, 7, 4, z);

    fclose(f); delete m;
    printf("Press any key to exit");
    getch();
    return;
}
//-----

```

## Пример 2

Посимвольное копирование файла, включая символы перевода каретки (13) и конца строки (10). Имена файлов переданы через параметры программы.

Файлы в этой программе могут быть открыты как текстовые (r или rt и w или wt) или как двоичные (rb или r+b и wb или w+b). t – text, b – binary, r – read, w – write

```

//-----
#include <stdio.h> // FILE fopen fclose fprintf, fscanf, printf, scanf
#include <conio.h> // getch
#include <windows.h> // SetConsoleOutputCP
//-----
void main(int argc, char *argv[])
{
    SetConsoleOutputCP(1251);
    printf("Use \"Lucida Console\\n"); // не забудьте в свойствах консольного окна указать шрифт

    if (argc!=3){
        printf("Error: Вы забыли передать имена файлов\nPress any key to continue...");
        getch();
        return;
    }

    FILE *fin, *fout;
    if ((fin = fopen( argv[1], "rt")) ==NULL) {

```

```

printf("Error: Ошибка при открытии файла %s для чтения\nPress any key", argv[1]);
getch(); // ожидание нажатия клавиши
return; // выход из функции main
}
if ((fout = fopen( argv[2], "wt")) ==NULL) {
printf("Error: Ошибка при открытии файла %s для записи\nPress any key", argv[2]);
fclose(fin);
getch(); // ожидание нажатия клавиши
return; // выход из функции main
}

char ch;
while (!feof(fin)){
ch = getc(fin);
if (!feof(fin)) putc(ch, fout);
}

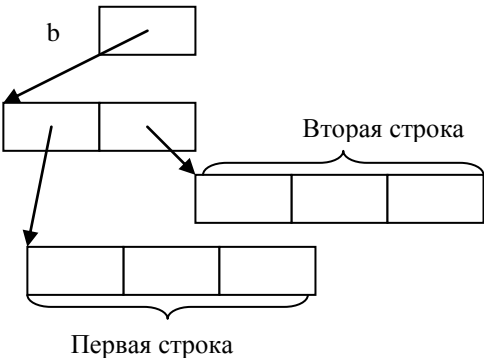
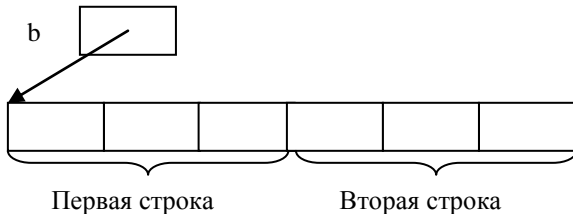
fclose(fin);
fclose(fout);
printf("THE END\nPress any key to exit");
getch();
return;
}
//-----

```

### Одномерный массив

	Динамический	Статический
Описание переменной	int a[] = {0,1,2,30,-40}; // с инициализацией int *b;	int a[5]; int b[3] = {0,1,2}
Выделение памяти	a = new int[5]; // с++ b = (int*)calloc(3, sizeof(int)); // с обнулением b = (int*)malloc(3*sizeof(int)); // без обнуления b = (int*)realloc(3, sizeof(int)); // перевыделение	При описании
Обращение к элементу	a[0]    *a b[i]    *(b+i)	a[0]    *a b[i]    *(b+i)
Освобождение памяти	delete [] a; // с++ free(b);	При выходе из области видимости, если не static
Описание параметра	Для изменения значений void proc1(int a[], int *b); void proc2(int *a, int b[]); void proc3(int a[5], int b[3]);	
	Для выделения/освобождения памяти void proc4(int *a[], int **b); void proc5(int **a, int *b[]); void* func_a(int a[]); // вызов a=(int*)func_a(a); int* func_b(int *b); // вызов b=func_b(b);	Невозможно освободить/ перевыделить память
Структура	Указатель на первый элемент (int a[5])  <div style="display: flex; align-items: center;"> <span style="margin-right: 10px;">a</span> <div style="border: 1px solid black; width: 80px; height: 20px; display: inline-block;"></div> <span style="font-size: 2em; margin: 0 10px;">→</span> <div style="border: 1px solid black; width: 400px; height: 20px; display: inline-block; position: relative;"> <div style="position: absolute; left: 0; right: 0; height: 20px;"></div> </div> </div>	

## Двухмерный массив

	Динамический (#include <alloc.h>)	Статический
Описание переменной	<code>int *a[] = {new int[3], new int[3]}; // 2x3 int **b, **c;</code>	<code>int a[5][3]; int b[2][3]={ {0,1,2},{3,4,5} };</code>
Выделение памяти	<code>c = new int*[5]; // 5x3 for(int i=0; i&lt;5; i++) c[i] = new int[3];  b = (int**)calloc(2, sizeof(int*)); // 2x3 for(int i=0; i&lt;2; i++)     b[i]= (int*)calloc(3, sizeof(int));</code>	При описании
Доступ к элементу	<code>a[0][0]       **a b[i][j]       *((int*)b+i)+j)</code>	<code>a[0][0]       **a b[i][j]       *((int*)b+i*3+j)</code>
Освобождение памяти	<code>for(i=0;i&lt;5;i++) delete [] a[i]; delete [] a;  for(int i=0;i&lt;2;i++) free(b[i]); free(b);</code>	При выходе из области видимости, если не static
Описание параметра	Для изменения значений <code>void proc1(int *a[], int **b); void proc2(int **a, int *b[]);</code>	Для изменения значений <code>void proc1(int a[5][3], int b[2][3]); void proc2(int *a[3], int b[][3]);</code>
	Для выделения/освобождения памяти <code>void proc3(int **a[], int ***b); void proc4(int ***a, int **b[]); void* func(int *a[]); //a=(int**)func(a);</code>	Невозможно освободить/ перевыделить память
Структура	<p>Указатель на массив указателей на первые элементы строк (2x3)</p> 	<p>Указатель на первый элемент Строки одна за другой (2x3)</p> 

## Функции и параметры

```

<тип_результата> <имя_функции>(<список_параметров>){
    <операторы>
    return <результат>;
}

```

Например, вычислим для целых чисел  $c = a + b$

$c$ – возвращаемое значение в точку вызова функции	
Кратко	Аналог на Delphi
<b>Способ1</b> <code>int func1(int a, int b){     return a+b; }</code>	<code>Function func1( a,b: integer):integer; Begin     func1:=a+b; exit; End;</code>

<b>Вызов:</b> c=func1(a,b);	<b>Вызов:</b> c:=func1(a,b);
Подробно	Аналог на Delphi
<b>Способ2</b> int func2(int a, int b){ int res; res=a+b; return res; } <b>Вызов:</b> c=func2(a,b);	Function func2( a,b: integer):integer; Var res: integer; Begin res:=a+b; func2:=res; exit; End; <b>Вызов:</b> c:=func2(a,b);
<b>Изменяем c через параметр</b>	
Передаем адрес для изменения значения по этому адресу	Аналог на Delphi (Type PInteger = ^Integer;)
<b>Способ3 (указатель)</b> void func3(int a, int b, int *c){ int res; res=a+b; *c=res; return; } <b>Вызов:</b> func3(a,b,&c); например, scanf("%d", &n);	Procedure func3( a,b: integer; c: PInteger); Var res: integer; Begin res:=a+b; c^:=res; exit; End; <b>Вызов:</b> func3(a,b,@c); // var a,b,c:integer;
Передача по ссылке (не во всех версиях компилятора Си!)	Аналог на Delphi
<b>Способ4 (ссылка)</b> void func4(int a, int b, int &c){ c=a+b; return; } <b>Вызов:</b> c=func4(a,b,c); <b>НЕ ИСПОЛЬЗОВАТЬ В ЛАБОРАТОРНЫХ РАБОТАХ ЭТОТ СПОСОБ – УЧИМ УКАЗАТЕЛИ ПО СПОСОБУ №3 – способ, использованный в <u>большинстве</u> библиотечных функций</b>	Procedure func4( a,b: integer; var c: integer); Begin c:=a+b; exit; End; <b>Вызов:</b> func4(a,b,c);

## Прототипы функций и разделение на файлы

Для возможности раздельной разработки и компиляции, для создания библиотек функций есть возможность создания многофайловых проектов.

Функции, выделяемые помимо главной функции приложения *main*, обычно располагают не до, а после *main*. Но, поскольку их описание должно быть сделано ДО использования, в начало файла помещают прототипы этих функций, например:

Функция ДО main	Функция ПОСЛЕ main	Комментарии
<pre>#include &lt;conio.h&gt; // getch #include &lt;stdio.h&gt; // printf, scanf //----- int sum(int a, int b){ return a+b; } //----- void main(){ int a,b; printf("a=? b=?"); scanf("%d %d", &amp;a, &amp;b);</pre>	<pre>#include &lt;conio.h&gt; // getch #include &lt;stdio.h&gt; // printf, scanf //----- int sum(int, int); //----- void main(){ int a,b; printf("a=? b=?"); scanf("%d %d", &amp;a, &amp;b);  int c=sum(a,b); printf("c=%d+%d=%d", a,b,c);</pre>	<p>Подключение библиотек</p> <p>Прототип <i>sum</i> (можно без имен параметров)</p> <p>Главная функция приложения <i>main</i></p> <p>Вызов функции <i>sum</i> и объявление <i>c</i></p>



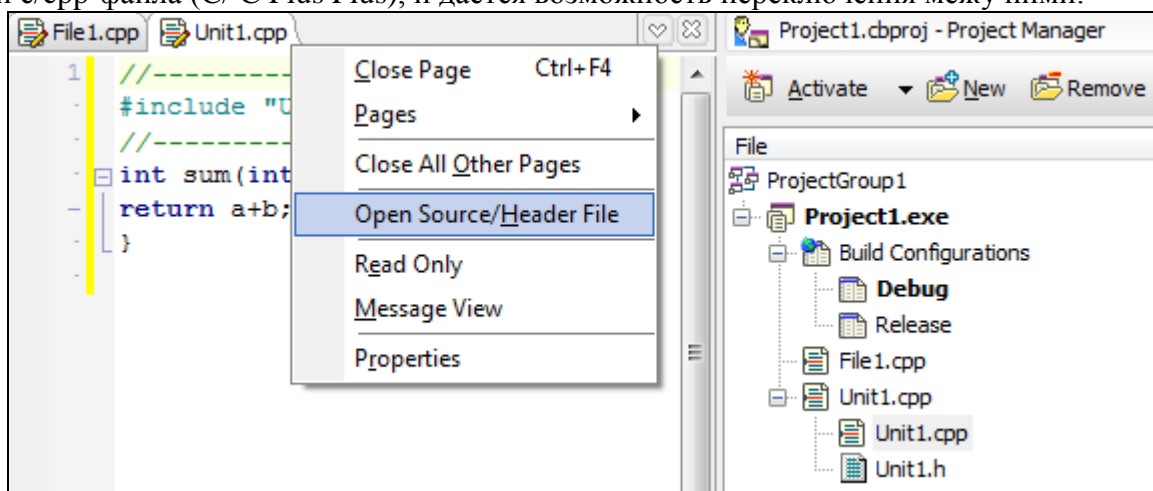
<pre>int c=sum(a,b); printf("c=%d+%d=%d", a,b,c);  printf("\nPress any key"); getch(); return; } //-----</pre>	<pre>printf("\nPress any key"); getch(); return; } //----- int sum(int a, int b){ return a+b; }</pre>	Функция <i>sum</i>
----------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------	--------------------

Описания прототипов и констант обычно выносят в заголовочные файлы (\*.h), которые имеют то же имя, что и файл с полным описанием этих функций (\*.cpp или \*.c), но другое расширение. Эти пары файлов имеют смысл аналогичный разделам INTERFACE и IMPLEMENTATION в модулях (Unit) в языке Delphi.

Если вынести функцию *sum* из предыдущего примера в отдельный файл, получится код:

Заголовочный файл <i>Unit1.h</i>	Файл с главной функцией <i>Main</i>	Комментарии
<pre>int sum(int, int);</pre>	<pre>#include &lt;conio.h&gt; // getch #include &lt;stdio.h&gt; // printf, scanf //----- #include "Unit1.h" // sum //-----  void main(){ int a,b; printf("a=? b=?"); scanf("%d %d", &amp;a, &amp;b);  int c=sum(a,b); printf("c=%d+%d=%d", a,b,c);  printf("\nPress any key"); getch(); return; }</pre>	Подключение стандартных библиотек (в угловых скобках)
<pre>Файл с функцией Unit1.cpp  //если в заголовочном файле //есть описания констант, типов, //то его надо подключить и //к этому файлу, как и другие //необходимые модули: //----- #include "Unit1.h" //----- int sum(int a, int b){ return a+b; }</pre>		Подключение своей библиотеки (путь в двойных кавычках)

При создании модуля (New→ Unit C++ Builder) в среде CodeGear RAD Studio2003-2009, как и в некоторых других средах сразу создается заготовка для двух файлов: заголовочного h-файла (h – header) и c/cpp-файла (C/ C Plus Plus), и дается возможность переключения между ними:



После раздельной компиляции исходного кода всех модулей создаются объектные файлы, которые после редактирования связей компоновщиком преобразуются в исполнимый код программы в виде *exe*-файла (либо в *dll*-файл, например, в зависимости от исходно выбранного типа проекта).