

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ИНФОРМАТИКА

1 Семестр

Раздел 1 Основы программирования на С, ветвления, циклы. (Темат.занятия 1-4)

1.1 Контроль по итогам (КИ) - 7 Неделя

Текущий контроль успеваемости заключается в проверке выполнения заданий на лабораторных работах, проверке домашних заданий (для самостоятельной работы) с учетом своевременности их выполнения. В конце освоения раздела проводится тестирование.

Контроль освоения раздела дисциплины проводится преподавателем в форме контроля по итогам (КИ). Баллы за каждый раздел дисциплины вычисляются путем суммирования баллов, полученных студентом, за каждое входящее в данный раздел мероприятие текущего контроля: задания для лабораторных работ (ЛР), задания для самостоятельных работ (ДЗ), тестовое задание (Т).

Критерии оценки заданий для лабораторных работ (ЛР) и заданий для самостоятельных работ (ДЗ).

Программа, составленная студентом в ходе выполнения задания, должна соответствовать следующим требованиям:

- выполнены все условия, указанные в задании;
- компиляция и компоновка проходят успешно;
- во время выполнения программы не происходят сбои и зависания;
- интерфейс ввода-вывода соответствует требованиям для автоматической проверки решений;
- автоматическая проверка решений выполняется полностью и успешно;
- текст исходного кода программы оформлен в соответствии с требованиями структурного программирования;
- программа содержит комментарий к самым важным фрагментам кода.

За невыполнение каждого из перечисленных требований оценка решения может быть снижена на 1 балл. Штрафные (отрицательные) баллы не начисляются.

Критерии оценки тестовых заданий (Т).

Тестирование студентов в конце освоения каждого раздела дисциплины проводится на занятиях в компьютерном классе с помощью электронной обучающей системы.

Тестовое задание считается выполненным успешно, если студент набрал не менее 60% от максимального балла (округление результата проводится по обычным правилам).

Аттестация студента по разделу.

Контроль по итогам (КИ) считается пройденным успешно, а студент аттестованным по разделу, если он набрал не менее 60% от максимально возможного балла. Для данного раздела аттестационный балл равен **14**.

Оценка уровня сформированности каждой компетенции может быть получена путем вычисления процентной доли суммы набранных студентом баллов от

максимально возможной суммы баллов за задания раздела, которые входят в перечень контрольных мероприятий соответствующей компетенции (округление проводится по обычным правилам).

1.2 Лабораторная работа (ЛР) - 3 Неделя

Задание для лабораторной работы 1.

Тема 1: Введение в язык С. Данные, операции, программы.

Составить программу, которая запрашивает у пользователя два целых числа и находит остаток от целочисленного деления произведения этих чисел на их сумму.

1.3 Домашнее задание (ДЗ) - 4 Неделя

Задание для самостоятельной работы 1.

Тема 1: Введение в язык С. Данные, операции, программы.

Составить программу, которая выполняет следующее.

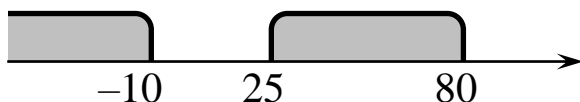
- 1) Определяет сумму цифр четырехзначного числа, введенного пользователем.
- 2) Запрашивает у пользователя целочисленные корни многочлена четвертой степени и, используя теорему Виета, находит его коэффициенты. Считать, что старший коэффициент многочлена равен единице. Вывести результат в форме таблицы, где первая строка – степени, а вторая – коэффициенты при этих степенях.

1.4 Лабораторная работа (ЛР) - 4 Неделя

Задание для лабораторной работы 2.

Тема 2: Логические выражения. Разветвляющиеся алгоритмы.

Составить программу, которая **проверяет четность** заданного пользователем целого числа (четное – «even» / нечетное – «odd»), если это число входит в указанный диапазон (если число не входит в диапазон, программа выводит сообщение «excluded»):



Четное число — целое число, которое делится на 2 без остатка: ..., -4, -2, 0, 2, 4, 6, 8, ...

Примечание: считать, что граничные значения интервалов входят внутрь интервалов.

1.5 Домашнее задание (ДЗ) - 5 Неделя

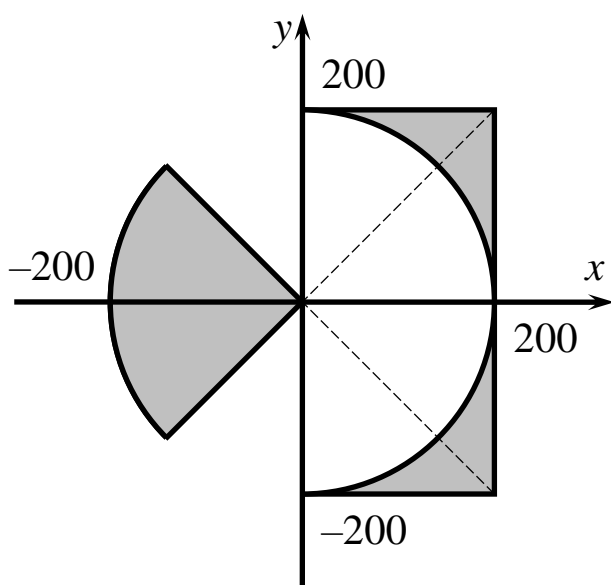
Задание для самостоятельной работы 2.

Тема 2: Логические выражения. Разветвляющиеся алгоритмы.

Составить программу, которая выполняет следующее.

1) Найти и вывести наибольшее из трёх целых чисел, введенных пользователем. Если два одинаковых числа больше третьего, вывести значение числа, пробел и сообщение «two». Если все числа равны, вывести только сообщение «equal».

2) Пользователь вводит целые числа x и y . Определить, принадлежит ли точка с координатами (x, y) заштрихованной области на рисунке, и вывести одно из сообщений: «inside», «outside».



Примечание: считать, что границы областей не входят внутрь областей.

1.6 Лабораторная работа (ЛР) - 5 Неделя

Задание для лабораторной работы 3.

Тема 3: Циклические алгоритмы.

Пользователь вводит целое положительное число m . Найти наименьшее натуральное число k , при котором $3^k > m$. Вывести на экран таблицу степеней числа 3 (от первой до 15-й). (Стандартную функцию возведения в степень использовать не следует.)

1.7 Домашнее задание (ДЗ) - 6 Неделя

Задание для самостоятельной работы 3.

Тема 3: Циклические алгоритмы.

Числа Фибоначчи определяются так: первое и второе числа равны 1, каждое следующее равно сумме двух предыдущих. Составить программу, которая находит первые k чисел Фибоначчи (k – задается пользователем).

1.8 Лабораторная работа (ЛР) - 6 Неделя

Задание для лабораторной работы 4.

Тема 4: Работа с циклическими конструкциями.

1) Найти количество делителей каждого из целых чисел от 190 до 220. (Единицу и само число учитывать в качестве делителей не следует.) Вывести результат в виде следующей таблицы (второй столбец содержит список делителей, разделенных запятыми, по четыре числа в одной строке):

304	2, 4, 8, 16, 19, 38, 76, 152	8
305	5, 61	2
306	2, 3, 6, 9, 17, 18, 34, 51, 102, 153	10
307		0
308	2, 4, 7, 11, 14, 22, 28, 44, 77, 154	10

Задачу следует решить без
использования массива и
вызова в файл

2) Определить наибольшее число из данного интервала, у которого количество делителей максимально.

1.9 Домашнее задание (ДЗ) - 7 Неделя

Задание для самостоятельной работы 4.

Тема 4: Работа с циклическими конструкциями.

1) Пользователь вводит целое положительное число k . Найти наибольшее значение суммы

$$1^k + 2^k + 3^k + 4^k + \dots + n^k,$$

которое не выходит за пределы диапазона типа long int. Определить значение n для найденной суммы.

2) Определить, какая операция является причиной переполнения при попытке вычисления значения суммы для $n+1$. Если причиной переполнения является суммирование слагаемых, то вывести символ «+» (плюс), если – умножение (при возведении в степень), то вывести символ «*» (звёздочка).

(В программе не следует использовать стандартную функцию возведения в степень, массивы и вычисления с плавающей точкой.)

1.10 Тестирование (Т) - 7 Неделя

Тестовое задание по разделу I.

Вопрос 1

Определить значение следующего выражения: $1 - 2 * - 3$

- 7
 - 3
 - -7
 - -4
 - -5
 - 5
-

Вопрос 2

Определить значение следующего выражения: $2 + 10 / 2 * 3$

- 17
 - 18
 - 3
 - 2
 - 21
 - 3.667
-

Вопрос 3

Укажите при каких значениях `int x`; следующее выражение истинно:
`x>5 || x>=3 && x<4`

- $x \in 3 \cup [6; \text{INT_MAX}]$
 - $x = 3$
 - $x \in [\text{INT_MIN}; 3] \cup [6; \text{INT_MAX}]$
 - $x \in [6; \text{INT_MAX}]$
 - $x \in [\text{INT_MIN}; 3]$
 - выражение всегда ложно
-

Вопрос 4

Определить что будет выведено на экран при `int x = 2`;

```
if (x<4 || x>6)
    if (x>2 && x<8)
        if (!(x<5))
            printf("1");
        else printf("2");
    else printf("3");
else printf("4");
```

- 3
 - 1
 - 2
 - 4
 - 13
 - не будет выведено ничего
-

Вопрос 5

Определить сколько раз будет выполнено тело цикла.

```
int i;  
for (i=7; i%2>=1; --i)  
{  
    ...  
}
```

- 1
 - 4
 - 3
 - 7
 - бесконечный цикл
 - не будет выполнено ни разу
-

Вопрос 6

Определить сколько раз будет выполнено тело цикла.

```
int i=1;  
while (i*i<64) {  
    ...  
    i*=2;  
}
```

- 3
 - 2
 - 1
 - 4
 - 6
 - не будет выполнено ни разу
-

Вопрос 7

Определить сколько раз будет выполнено тело цикла.

```
int i=0;  
do {  
    ++i;  
    ...  
} while (1/i>0);
```

- 2
 - 1
 - 3
 - 4
 - бесконечный цикл
 - не будет выполнено ни разу
-

Вопрос 8

Определить значение переменной **x** после выполнения фрагмента кода:

```
int x=0, y=0;  
while (y<10)  
    y=3*(++x)+1;
```

- 3
 - 4
 - 2
 - 1
 - 5
 - бесконечный цикл
-

Раздел 2 Вычисления, массивы и указатели. (Темат.занятия 5-7)

2.1 Контроль по итогам (КИ) - 12 Неделя

Текущий контроль успеваемости заключается в проверке выполнения заданий на лабораторных работах, проверке домашних заданий (для самостоятельной работы) с учетом своевременности их выполнения. В конце освоения раздела проводится тестирование.

Контроль освоения раздела дисциплины проводится преподавателем в форме контроля по итогам (КИ). Баллы за каждый раздел дисциплины вычисляются путем суммирования баллов, полученных студентом, за каждое входящее в данный раздел мероприятие текущего контроля: задания для лабораторных работ (ЛР), задания для самостоятельных работ (ДЗ), тестовое задание (Т).

Критерии оценки заданий для лабораторных работ (ЛР) и заданий для самостоятельных работ (ДЗ).

Программа, составленная студентом в ходе выполнения задания, должна соответствовать следующим требованиям:

- выполнены все условия, указанные в задании;
- компиляция и компоновка проходят успешно;
- во время выполнения программы не происходят сбои и зависания;
- интерфейс ввода-вывода соответствует требованиям для автоматической проверки решений;
- автоматическая проверка решений выполняется полностью и успешно;
- текст исходного кода программы оформлен в соответствии с требованиями структурного программирования;
- программа содержит комментарий к самым важным фрагментам кода.

За невыполнение каждого из перечисленных требований оценка решения может быть снижена на 1 балл. Штрафные (отрицательные) баллы не начисляются.

Критерии оценки тестовых заданий (Т).

Тестирование студентов в конце освоения каждого раздела дисциплины проводится на занятиях в компьютерном классе с помощью электронной обучающей системы.

Тестовое задание считается выполненным успешно, если студент набрал не менее 60% от максимального балла (округление результата проводится по обычным правилам).

Аттестация студента по разделу.

Контроль по итогам (КИ) считается пройденным успешно, а студент аттестованным по разделу, если он набрал не менее 60% от максимально возможного балла. Для данного раздела аттестационный балл равен **17**.

Оценка уровня сформированности каждой компетенции может быть получена путем вычисления процентной доли суммы набранных студентом баллов от максимально возможной суммы баллов за задания раздела, которые входят в перечень контрольных мероприятий соответствующей компетенции (округление проводится по обычным правилам).

2.2 Лабораторная работа (ЛР) - 9 Неделя

Задание для лабораторной работы 5.

Тема 5: Вычисления с плавающей точкой.

Составить программу, которая для различных значений целого числа N из интервала от 20 000 000 до 200 000 000 (с шагом 20 000 000, включая границы интервала) вычисляет сумму

$$\sum_{n=1}^N \sqrt{\frac{1+3n}{3+n}}$$

двумя способами: напрямую и с коррекцией (методом Кохена). Для вещественных чисел использовать переменные типа double.

Вывести на экран таблицу для сравнения результатов, которая должна содержать следующие значения:

- значение числа N ;
- сумма, вычисленная напрямую (без коррекции);
- сумма, вычисленная с коррекцией (методом Кохена);
- разность значений суммы без коррекции и суммы с коррекцией.

2.3 Домашнее задание (ДЗ) - 10 Неделя

Задание для самостоятельной работы 5.

Тема 5: Вычисления с плавающей точкой.

Известно следующее соотношение для суммы бесконечного ряда (при $n \rightarrow \infty$)

$$\frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \frac{1}{3 \cdot 4} + \dots + \frac{1}{n \cdot (n+1)} + \dots = 1$$

Составить программу, которая вычисляет значение суммы ряда, стоящего в левой части данного уравнения. Вычисляемое значение должно приближаться к аналитическому решению, стоящему в правой части уравнения.

Определить абсолютную погрешность вычисления суммы для заданного пользователем натурального числа n .

2.4 Лабораторная работа (ЛР) - 10 Неделя

Задание для лабораторной работы 6.

Тема 6: Реализация вычислительных методов на компьютере.

Составить программу, которая находит решение x_0 уравнения

$$x^2 - 6x + 4 = 0 \quad \text{на отрезке } x \in [3;8]$$

методом деления пополам. Определить x_0 с абсолютной погрешностью 1×10^{-10} .

2.5 Домашнее задание (ДЗ) - 11 Неделя

Задание для самостоятельной работы 6.

Тема 6: Реализация вычислительных методов на компьютере.

1) Составить программу, которая находит решение x_0 уравнения

$$\sin x = \frac{2}{5} \quad \text{на отрезке } x \in [2;4]$$

методом деления пополам. Определить x_0 с точностью 1×10^{-7} .

2) Используя найденное значение x_0 , вычислить значение $\sin x_0$ по формуле разложения функции в степенной ряд:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

с абсолютной погрешностью 1×10^{-6} .

(Стандартную функцию возведения в степень использовать не следует.)

2.6 Лабораторная работа (ЛР) - 11 Неделя

Задание для лабораторной работы 7.

Тема 7: Массивы и указатели.

Заполнить массив значениями функции $y = 5 \sin(2x)$ для всех значений x от -6 до 6 (включительно), взятых с шагом $0,3$. Найти максимальный элемент массива и среднее арифметическое всех элементов.

2.7 Домашнее задание (ДЗ) - 12 Неделя

Задание для самостоятельной работы 7.

Тема 7: Массивы и указатели.

1) Составить программу, которая заполняет массив случайными целыми числами из диапазона $[1; 20]$. Вывести массив на экран в виде таблицы (индекс, элемент).

2) Найти среднее арифметическое всех элементов массива, значение которых больше целого числа a , задаваемого пользователем. Реализовать данную часть задания двумя способами: а) с использованием индексов, б) при помощи указателей и адресной арифметики.

2.8 Тестирование (Т) - 11 Неделя

Тестовое задание по разделу II.

Вопрос 1

Число **0,75** представлено как число с плавающей запятой в виде $m \cdot 2^p$. Указать, чему равна дробная часть мантиисы m , нормализованной по правилу $1 \leq m < 2$.

- 10000...0
 - 11000...0
 - 01000...0
 - 01100...0
 - 00100...0
 - 10100...0
-

Вопрос 2

Выбрать наибольшую из перечисленных констант.

- 7e-3
 - 0.85e-3
 - 8E-4
 - 0.6E-2
 - 70E-5
 - 65e-4
-

Вопрос 3

Пусть из-за конечного количества разрядов компьютер не позволяет различать числа с плавающей точкой порядка 10^{30} . Тогда значение выражения

$$1.0e30 + 2.0 - 1.0e30 - 1.0$$

после выполнения всех операций станет равным

- -1.0
 - 1.0
 - 2.0
 - 0.0
 - 1.0e30
 - -1.0e30
-

Вопрос 4

Программа находит численное решение уравнения $\sin(x) = 0$ с помощью алгоритма, реализующего метод бисекции, на отрезке $x \in [2,4; 3,6]$ с точностью $\epsilon = 0,1$.

Указать, сколько итераций алгоритма при этом выполняется (на каждой итерации отрезок делится пополам один раз).

- 4
 - 1
 - 2
 - 3
 - 5
 - 6
-

Вопрос 5

Объявлен массив `int a[6];` Указать допустимые значения целочисленной переменной `i` в операторе

```
a[i] = 7;
```

- $i \in [0; 5]$
 - $i \in [0; 6]$
 - $i \in [1; 6]$
 - $i \in [1; 5]$
 - $i \in [0; 7]$
 - $i \in [1; 7]$
-

Вопрос 6

Объявлен массив

```
int a[] = {1, 2, 3, 4};
```

Указать допустимые значения целочисленной переменной `i` в операторе

```
a[i] = 5;
```

- $i \in [0; 3]$
 - $i \in [0; 4]$
 - $i \in [1; 4]$
 - $i \in [1; 3]$
 - $i \in [0; 5]$
 - $i \in [1; 5]$
-

Вопрос 7

Объявлен массив `int a[6];` Указать допустимые значения целочисленной переменной `i` в операторе

```
*(a+i) = 7;
```

- $i \in [0; 5]$
 - $i \in [0; 6]$
 - $i \in [1; 6]$
 - $i \in [1; 5]$
 - $i \in [0; 7]$
 - $i \in [1; 7]$
-

Вопрос 8

Для объявлений

```
int a[10], *p;
```

указать номера допустимых операторов, при выполнении которых не происходит выхода за границы массива:

```
a = p;      /* 1 */
p = a+5;    /* 2 */
a = p+5;    /* 3 */
p = a+10;   /* 4 */
```

- только 2
 - 2 и 4
 - только 1
 - никакие не допустимы
 - 1 и 3
 - все операторы допустимы
-

Вопрос 9

Сделаны следующие объявления: `int i, m[5];`

Определить, что будет выведено на экран в результате выполнения фрагмента кода:

```
for (i=0; i<5; ++i)
    m[i]=2*(i+1);
for (i=0; i<4; ++i)
    printf("%d ", m[i]+m[i+1]);
```

- 6 10 14 18
 - 6 10 14 18 22
 - 2 4 6 8
 - 2 4 6 8 10
 - 3 5 7 9
 - 3 5 7 9 11
-

Раздел 3 Функции, передача параметров. рекурсия. (Темат.занятия 8-10)

3.1 Контроль по итогам (КИ) - 16 Неделя

Текущий контроль успеваемости заключается в проверке выполнения заданий на лабораторных работах, проверке домашних заданий (для самостоятельной работы) с учетом своевременности их выполнения. В конце освоения раздела проводится тестирование.

Контроль освоения раздела дисциплины проводится преподавателем в форме контроля по итогам (КИ). Баллы за каждый раздел дисциплины вычисляются путем суммирования баллов, полученных студентом, за каждое входящее в данный раздел мероприятие текущего контроля: задания для лабораторных работ (ЛР), задания для самостоятельных работ (ДЗ), тестовое задание (Т).

Критерии оценки заданий для лабораторных работ (ЛР) и заданий для самостоятельных работ (ДЗ).

Программа, составленная студентом в ходе выполнения задания, должна соответствовать следующим требованиям:

- выполнены все условия, указанные в задании;
- компиляция и компоновка проходят успешно;
- во время выполнения программы не происходят сбои и зависания;
- интерфейс ввода-вывода соответствует требованиям для автоматической проверки решений;
- автоматическая проверка решений выполняется полностью и успешно;
- текст исходного кода программы оформлен в соответствии с требованиями структурного программирования;
- программа содержит комментарий к самым важным фрагментам кода.

За невыполнение каждого из перечисленных требований оценка решения может быть снижена на 1 балл. Штрафные (отрицательные) баллы не начисляются.

Критерии оценки тестовых заданий (Т).

Тестирование студентов в конце освоения каждого раздела дисциплины проводится на занятиях в компьютерном классе с помощью электронной обучающей системы.

Тестовое задание считается выполненным успешно, если студент набрал не менее 60% от максимального балла (округление результата проводится по обычным правилам).

Аттестация студента по разделу.

Контроль по итогам (КИ) считается пройденным успешно, а студент аттестованным по разделу, если он набрал не менее 60% от максимально возможного балла. Для данного раздела аттестационный балл равен **17**.

Оценка уровня сформированности каждой компетенции может быть получена путем вычисления процентной доли суммы набранных студентом баллов от максимально возможной суммы баллов за задания раздела, которые входят в перечень контрольных мероприятий соответствующей компетенции (округление проводится по обычным правилам).

3.2 Лабораторная работа (ЛР) - 13 Неделя

Задание для лабораторной работы 8.

Тема 8: Функции.

Известны значения радиуса основания (r) и высоты (h) **цилиндра**.

- 1) Составить функцию, которая получает эти значения в качестве параметров и возвращает значение **площади полной поверхности** (S) данного тела. Составить такую же функцию для вычисления его **объема** (V).
- 2) Составить функцию, которая по заданному объему **шара** находит значение его радиуса. И с ее помощью вычислить радиус шара, объем которого равен объему заданного пользователем цилиндра.

3.3 Домашнее задание (ДЗ) - 14 Неделя

Задание для самостоятельной работы 8.

Тема 8: Функции.

Пользователь вводит три целых числа: a, b, c . Составить функцию для решения квадратного уравнения $ax^2 + bx + c = 0$, которая получает введенные числа в качестве параметров. Найденные корни уравнения функция возвращает в качестве двух других параметров.

Результат, который возвращает сама функция, должен позволять отслеживать следующие случаи (зависящие от значений введенных чисел):

- 1) уравнение не квадратное (вырожденное),
- 2) нет вещественных корней,
- 3) корни одинаковые (при этом уравнение – квадратное).

3.4 Лабораторная работа (ЛР) - 14 Неделя

Задание для лабораторной работы 9.

Тема 9: Функции: передача параметров.

Составить программу для работы с целыми положительными числами в **троичной** системе счисления.

Для хранения таких чисел использовать стандартный тип `unsigned long int`, но ограничиться только цифрами 0,1,2. Например, для переменной k в троичной системе счисления `unsigned long int k=120`; значение будет равно 15 в десятичной системе счисления. Учесть, что при таком представлении можно работать только с числами, ограниченными диапазоном типа `unsigned long int`. В качестве диапазона чисел в троичной системе счисления можно взять значения от нуля до 2 222 222 222.

1) Составить функцию для перевода таких чисел в десятичную систему счисления. Функция должна получать исходное число в форме параметра и возвращать результат.

2) Составить функцию для перевода чисел из десятичной системы счисления в троичную. Функция должна получать исходное число и возвращать результат в форме параметров, а возвращаемое самой функцией значение должно отслеживать переполнение.

3.5 Домашнее задание (ДЗ) - 15 Неделя

Задание для самостоятельной работы 9.

Тема 9: Функции: передача параметров.

В соответствии с условиями задания для лабораторной работы к данному занятию составить функции для реализации следующих арифметических операций в указанной системе счисления: *сложение, вычитание, умножение*. Функции должны отслеживать возможное переполнение.

Для чисел a и b , введенных пользователем в указанной системе счисления, с помощью реализованных функций вычислить значение выражения $a(b-a)+b$.

3.6 Лабораторная работа (ЛР) - 15 Неделя

Задание для лабораторной работы 10.

Тема 10: Рекурсия.

Составить программу, которая, используя **рекурсивную функцию**, находит значение данной функции для любых целых неотрицательных аргументов:

$$F(n, m) = \begin{cases} 1, & \text{если } n = 0; \\ -1, & \text{если } n < m; \\ 2 \cdot F(n-1, m) & \text{в остальных случаях.} \end{cases}$$

3.7 Домашнее задание (ДЗ) - 16 Неделя

Задание для самостоятельной работы 10.

Тема 10: Рекурсия.

Составить рекурсивную функцию, которая находит k -е число последовательности, описанной в *задании для самостоятельной работы 3 (к тематическому занятию 3)*, значение k задается пользователем.

Вывести на экран **таблицу трассировки** значений переменных, используемых для рекурсивных вычислений. Обязательные поля таблицы: *текущий уровень рекурсии, значения на рекурсивном спуске, значения на рекурсивном возврате*. Таблица может быть разделена на две части: рекурсивный спуск (от первого уровня к последнему), рекурсивный возврат (от последнего уровня к первому).

3.8 Тестирование (Т) - 16 Неделя

Тестовое задание по разделу III.

Вопрос 1

При объявлении функции описан ее заголовок:

```
int func(int a, int b) {  
    ...  
}
```

Определить, какой прототип соответствует этой функции:

- `int func(int, int);`
 - `int func(int*, int*);`
 - `int func(*int, *int);`
 - `int func(a, b);`
 - `int func(a*, b*);`
 - `int func(*a, *b);`
-

Вопрос 2

В программе объявлены переменные:

```
double x;  
char c;
```

Значения обеих переменных не определены.

Объявлена функция:

```
void func(double *a, char b) {  
    ...  
}
```

Определить, какой вызов функции не содержит ошибок в коде программы и логике ее работы:

- `func(&x, 'c');`
 - `func(1.23, 'c');`
 - `func(1.23, c);`
 - `func(&x, c);`
 - `func(x, c);`
 - `func(x, 'c');`
-

Вопрос 3

Объявлена функция:

```
int f(int x1, int x2) {  
    if (x1%2)  
        return x2;  
    else  
        return x1;  
}
```

Определить, какое значение примет целочисленная переменная **x** в результате выполнения фрагмента кода:

```
int a=2, b=3;  
x = f(a,b) + f(a+b,b-a);
```

- 3
 - 4
 - 7
 - 8
 - 1
 - Невозможно определить из-за побочных эффектов.
-

Вопрос 4

Объявлена функция:

```
void func(int p1, int *p2) {  
    p1++;  
    (*p2)++;  
    p2++;  
}
```

Определить, какое значение примет целочисленная переменная **x** в результате выполнения фрагмента кода:

```
int a=1, b=1;  
func(a,&b);  
x = a+2*b;
```

- 5
 - 3
 - 4
 - 6
 - 7
 - 8
-

Вопрос 5

Определить, что будет выведено на экран в результате выполнения программы:

```
#include <stdio.h>
void f1(int);
void f2(int);
int main(void){
    int x=1;
    f1(x);
    printf("%d", x);
    return 0;
}
void f1(int x){
    x*=2;
    f2(x);
}
void f2(int x){
    x*=3;
}
```

- 1
 - 2
 - 3
 - 6
 - 12
 - Ошибка при компиляции.
-

Вопрос 6

Определить, что будет выведено на экран в результате выполнения программы:

```
#include <stdio.h>
int func(int a, int *b){
    *b=0;
    printf("%d", a);
    if (a) return 0;
    else return 1;
}
int main(void){
    int x=1;
    if (func(x,&x)) x=1;
    else printf("%d", x);
    return 0;
}
```

- 10
 - 0
 - 01
 - 11
 - 1
 - 00
-

Вопрос 7

Определить значение целочисленной переменной **x** в результате выполнения функции:

```
int f(int k) {  
    if (k>0)  
        return 3*f(k-1);  
    else  
        return 1;  
}
```

вызванной следующим образом:

```
x = f(4);
```

- 81
 - 243
 - 27
 - 12
 - 64
 - 4
-

Вопрос 8

Определить значение целочисленной переменной **x** в результате выполнения функции:

```
int f(int a){  
    a=1+f(a-1);  
    if (a<=0) return 0;  
    else return a;  
}
```

вызванной следующим образом:

```
x = f(3);
```

- Бесконечный рекурсивный вызов.
 - 3
 - 2
 - 1
 - 0
 - 4
-

2 Семестр

Раздел 1 Функции и массивы. Файлы. (Темат.занятия 11-13)

1.1 Контроль по итогам (КИ) - 7 Неделя

Текущий контроль успеваемости заключается в проверке выполнения заданий на лабораторных работах, проверке домашних заданий (для самостоятельной работы) с учетом своевременности их выполнения. В конце освоения раздела проводится тестирование.

Контроль освоения раздела дисциплины проводится преподавателем в форме контроля по итогам (КИ). Баллы за каждый раздел дисциплины вычисляются путем суммирования баллов, полученных студентом, за каждое входящее в данный раздел мероприятие текущего контроля: задания для лабораторных работ (ЛР), задания для самостоятельных работ (ДЗ), тестовое задание (Т).

Критерии оценки заданий для лабораторных работ (ЛР) и заданий для самостоятельных работ (ДЗ).

Программа, составленная студентом в ходе выполнения задания, должна соответствовать следующим требованиям:

- выполнены все условия, указанные в задании;
- компиляция и компоновка проходят успешно;
- во время выполнения программы не происходят сбои и зависания;
- интерфейс ввода-вывода соответствует требованиям для автоматической проверки решений;
- автоматическая проверка решений выполняется полностью и успешно;
- текст исходного кода программы оформлен в соответствии с требованиями структурного программирования;
- программа содержит комментарий к самым важным фрагментам кода.

За невыполнение каждого из перечисленных требований оценка решения может быть снижена на 1 балл. Штрафные (отрицательные) баллы не начисляются.

Критерии оценки тестовых заданий (Т).

Тестирование студентов в конце освоения каждого раздела дисциплины проводится на занятиях в компьютерном классе с помощью электронной обучающей системы.

Тестовое задание считается выполненным успешно, если студент набрал не менее 60% от максимального балла (округление результата проводится по обычным правилам).

Аттестация студента по разделу.

Контроль по итогам (КИ) считается пройденным успешно, а студент аттестованным по разделу, если он набрал не менее 60% от максимально возможного балла. Для данного раздела аттестационный балл равен **17**.

Оценка уровня сформированности каждой компетенции может быть получена путем вычисления процентной доли суммы набранных студентом баллов от максимально возможной суммы баллов за задания раздела, которые входят в перечень

контрольных мероприятий соответствующей компетенции (округление проводится по обычным правилам).

1.2 Лабораторная работа (ЛР) - 1 Неделя

Задание для лабораторной работы 11.

Тема 11: Функции: работа с массивом.

Составить две функции для работы с массивом:

- 1) Составить функцию, которая заполняет массив целыми числами по формуле

$$m[i] = a \cdot (i + b),$$

где a и b – целые числа, которые передаются в функцию в качестве параметров.

- 2) Составить функцию, которая выводит массив на экран в виде двух строк: в первой – номера, во второй – значения элементов массива.

Массив и количество элементов в нем передаются в каждую из функций в качестве параметров. (Таким образом, каждая из этих функций может работать с любым массивом, переданным в нее через параметры.)

Создать два локальных массива (в функции `main`). Количество элементов первого массива – 8, второго – 12. С помощью составленных функций заполнить оба массива и вывести их на экран.

1.3 Домашнее задание (ДЗ) - 3 Неделя

Задание для самостоятельной работы 11.

Тема 11: Функции: работа с массивом.

Составить программу для работы с целочисленными массивами с помощью следующих функций (массив и количество элементов в нем передаются в каждую из функций в качестве параметров):

- 1) Обнуление элементов массива.
- 2) Заполнение массива значениями, вводимыми с клавиатуры.
- 3) Заполнение массива случайными числами из диапазона, заданного пользователем (границы диапазона передается в функцию в качестве параметров).
- 4) Вывод массива на экран.
- 5) Определение суммы элементов массива.

Создать в функции `main` три локальных массива, состоящих из 5, 6 и 8 элементов, и заполнить их целыми значениями из интервала $[10; 15]$.

Найти все значения, которые встречаются в каждом из трех массивов. Для этого создать внешний массив из 6-и элементов (в соответствии с количеством целых чисел в заданном интервале). Элементами внешнего массива будут значения, определяющие, в скольких массивах встречается каждое целое число из заданного интервала.

1.4 Лабораторная работа (ЛР) - 4 Неделя

Задание для лабораторной работы 12

Тема 12: Символы и строки. Работа с файлами.

Составить программу, которая выделяет отдельные слова из введенной пользователем строки. *Словом* считается последовательность букв латинского алфавита. Слова разделяются одним или несколькими пробельными символами (пробел, табуляция).

Вывести на экран все слова (каждое слово – в отдельной строке).

Найти самое длинное слово.

1.5 Домашнее задание (ДЗ) - 6 Неделя

Задание для самостоятельной работы 12

Тема 12: Символы и строки. Работа с файлами.

Составить программу, которая создает текстовый файл, состоящий из случайного количества (не более 30-и) непустых строк. В каждой строке файла содержится случайное количество (не более 20-и) случайных целых положительных чисел, разделенных пробелами.

Найти в файле номер строки, содержащей максимальное число (если таких строк несколько, то найти все).

Примечание:

Для преобразования строки в число можно использовать функцию `atoi()`, описанную в заголовочном файле стандартной библиотеки `<stdlib.h>`. Пример:

```
int i;  
char s[6] = "12345";  
i = atoi(s);
```

1.6 Лабораторная работа (ЛР) - 6 Неделя

Задание для лабораторной работы 13

Тема 13: Динамический массив: работа с элементами.

Создать локальный динамический массив (в функции `main`) из n целых чисел (число n вводит пользователь) и заполнить его с клавиатуры целыми числами из диапазона $[5; 25]$. (Для отладки программы следует заполнять массив случайными целыми числами из заданного диапазона.)

Удалить k -й элемент массива (со сдвигом), изменяя размер занимаемого массивом блока памяти (число k вводит пользователь).

Составить функцию, которая меняет местами первый и i -й элементы массива (число i вводит пользователь). При этом сам массив, текущее количество элементов в нем, а также введенное пользователем число i передаются в функцию в качестве параметров. Вызвать составленную функцию.

Вывести на экран исходный массив и массив, полученный после применения этих двух преобразований. Выводить массив на экран следует отдельной функцией.

1.7 Домашнее задание (ДЗ) - 7 Неделя

Задание для самостоятельной работы 13

Тема 13: Динамический массив: работа с элементами.

Дополнить действия над массивом, описанные в задании для самостоятельной работы к тематическому занятию 11, следующими функциями.

- 6) Сохранение массива в текстовый файл.
- 7) Чтение массива из текстового файла.
- 8) Удаление нескольких элементов из указанного места (со сдвигом).
- 9) Добавление нескольких элементов в указанное место (со сдвигом).

Функции, описанные в пунктах 7 и 8, должны изменять размер занимаемого массивом блока памяти. Для этого необходимо изменить решение задания для самостоятельной работы к тематическому занятию 11 так, чтобы исходный массив, с которым работают все функции, был динамическим (объявленным локально в функции main). Все функции должны работать с массивами как с параметрами.

1.8 Тестирование (Т) - 7 Неделя

Тестовое задание по разделу IV.

Вопрос 1

Массив `int m[5]`; заполнен числами. Определить, какой элемент массива изменит свое значение в результате выполнения функции

```
void func(int *a) {  
    *(a-1) += 3;  
}
```

вызванной следующим образом:

```
func(m+2);
```

- `m[1]`
 - `m[0]`
 - `m[2]`
 - `m[3]`
 - `m[4]`
 - Элементы массива не изменятся.
-

Вопрос 2

В программе объявлена функция:

```
void func(int a[]) {  
    int tmp;  
    tmp=*(a-1);  
    *(a-1)=a[2];  
    a[2]=tmp;  
}  
  
int main(void) {  
    int m[8]={1,2,3,4,5,6,7,8};  
    func(m+3);  
}
```

Определить, какие элементы массива `m` поменяются местами в результате выполнения функции.

- `m[2]` и `m[5]`
 - `m[2]` и `m[4]`
 - `m[2]` и `m[6]`
 - `m[1]` и `m[4]`
 - `m[1]` и `m[3]`
 - `m[1]` и `m[5]`
-

Вопрос 3

В программе объявлена функция:

```
int func(int *p, int *q) {  
    int s=0;  
    while (p<q)  
        s+=*(++p);  
    return s;  
}  
  
int main(void) {  
    int sum, m[6]={1,2,3,4,5,6};  
    sum=func(m+1,m+4);  
}
```

Определить значение переменной `sum` после выполнения функции.

- 12
 - 18
 - 9
 - 14
 - 20
 - 7
-

Вопрос 4

На компьютере используется кодировка ASCII. Объявлена переменная `char ch = 'm';`

Определить, что будет выведено на экран в результате выполнения оператора:

```
printf(" %d %o %x ", ch, ch, ch);
```

- 109 155 6d
 - 109 6d 155
 - 155 109 6d
 - 155 6d 109
 - 6d 109 155
 - 6d 155 109
-

Вопрос 5

Объявлена переменная `char ch = '\011';`

Определить, что будет выведено на экран в результате выполнения оператора:

```
printf(" %o %d ", ch, ch);
```

- 11 9
 - 011 9
 - 11 3
 - 3 011
 - 3 11
 - 9 011
-

Вопрос 6

Объявлены переменные

```
char s[7] = "define";
```

```
char *ps;
```

Определить, какой из операторов присваивания может быть вызван сразу после объявления переменных (и составлен без ошибок):

- `ps = &s[2];`
 - `*ps = s[2];`
 - `ps = s[2];`
 - `*ps = &s[2];`
 - `*ps = *s[2];`
 - `ps = *s[2];`
-

Вопрос 7

В текстовом потоке ввода содержится последовательность символов:

SENSE

Определить, сколько символов будет считано из потока в результате выполнения следующего фрагмента кода:

```
char c;  
c = getchar();  
while (c != getchar())  
    putchar(c);
```

- 4
 - 2
 - 1
 - 3
 - 5
 - 6
-

Вопрос 8

Определить, сколько символов будет помещено в текстовый поток вывода в результате выполнения следующего фрагмента кода:

```
char *p = "def";  
while (*p != '\f')  
    putchar(*p++);  
puts(p);
```

- 4
 - 6
 - 2
 - 7
 - 3
 - ни одного
-

Вопрос 9

В программе объявлена функция:

```
void func(int **a, int n) {  
    ...  
}
```

В функцию передан динамический массив из 10-и элементов.

Выбрать выражение, которое предоставляет функции доступ к значению 7-го элемента массива.

- (*a)[6]
 - *a[6]
 - (&a)[6]
 - &a[6]
 - *((&a)[6])
 - *(&a[6])
-

Вопрос 10

В программе объявлена функция:

```
void func(int **a, int n) {  
    . . .  
}
```

В функцию передан динамический массив из 10-и элементов.

Выбрать выражение, которое внутри функции является адресом 6-го элемента массива.

- *a+5
 - *(a+5)
 - &a+5
 - &(a+5)
 - &(*a+5)
 - &(* (a+5))
-

Раздел 2 Сортировка массивов. (Темат.занятия 14-15)

2.1 Контроль по итогам (КИ) - 11 Неделя

Текущий контроль успеваемости заключается в проверке выполнения заданий на лабораторных работах, проверке домашних заданий (для самостоятельной работы) с учетом своевременности их выполнения. В конце освоения раздела проводится тестирование.

Контроль освоения раздела дисциплины проводится преподавателем в форме контроля по итогам (КИ). Баллы за каждый раздел дисциплины вычисляются путем суммирования баллов, полученных студентом, за каждое входящее в данный раздел мероприятие текущего контроля: задания для лабораторных работ (ЛР), задания для самостоятельных работ (ДЗ), тестовое задание (Т).

Критерии оценки заданий для лабораторных работ (ЛР) и заданий для самостоятельных работ (ДЗ).

Программа, составленная студентом в ходе выполнения задания, должна соответствовать следующим требованиям:

- выполнены все условия, указанные в задании;
- компиляция и компоновка проходят успешно;
- во время выполнения программы не происходят сбои и зависания;
- интерфейс ввода-вывода соответствует требованиям для автоматической проверки решений;
- автоматическая проверка решений выполняется полностью и успешно;
- текст исходного кода программы оформлен в соответствии с требованиями структурного программирования;
- программа содержит комментарий к самым важным фрагментам кода.

За невыполнение каждого из перечисленных требований оценка решения может быть снижена на 1 балл. Штрафные (отрицательные) баллы не начисляются.

Критерии оценки тестовых заданий (Т).

Тестирование студентов в конце освоения каждого раздела дисциплины проводится на занятиях в компьютерном классе с помощью электронной обучающей системы.

Тестовое задание считается выполненным успешно, если студент набрал не менее 60% от максимального балла (округление результата проводится по обычным правилам).

Аттестация студента по разделу.

Контроль по итогам (КИ) считается пройденным успешно, а студент аттестованным по разделу, если он набрал не менее 60% от максимально возможного балла. Для данного раздела аттестационный балл равен **14**.

Оценка уровня сформированности каждой компетенции может быть получена путем вычисления процентной доли суммы набранных студентом баллов от максимально возможной суммы баллов за задания раздела, которые входят в перечень контрольных мероприятий соответствующей компетенции (округление проводится по обычным правилам).

2.2 Лабораторная работа (ЛР) - 8 Неделя

Задание для лабораторной работы 14

Тема 14: Сортировка массива.

Составить программу, которая для массива, заполненного целыми числами, проводит сортировку по неубыванию методом обмена («пузырька»).

Выводить массив на экран следует отдельной функцией. (Отлаживать программу можно на массивах, заполненных случайными числами.)

2.3 Домашнее задание (ДЗ) - 9 Неделя

Задание для самостоятельной работы 14

Тема 14: Сортировка массива.

1) Дополнить программу работы с массивом, которая описана в заданиях для самостоятельной работы к занятиям 11 и 13, функцией сортировки массива методом, указанным в задании к лабораторной работе 14. (Функция должна работать с массивами как с параметрами.)

2) Модифицировать данный метод сортировки по принципу, повышающему эффективность алгоритма, описанному в Методических указаниях к занятию 14. Реализовать модифицированный (улучшенный) метод в отдельной функции.

2.4 Лабораторная работа (ЛР) - 10 Неделя

Задание для лабораторной работы 15

Тема 15: Усовершенствованные методы сортировки.

Изучить метод быстрой сортировки (Хоара). На основе алгоритма быстрой сортировки, реализованного в «Методических указаниях», составить программу, которая проводит сортировку массива целых чисел. Условие упорядоченности (неубывание/невозрастание) – то же, что и в задании для лабораторной работы к Тематическому занятию 14.

При каждой перестановке местами двух элементов массива, которые имеют разные индексы, выводить на экран весь массив целиком.

Выводить массив на экран следует отдельной функцией. (Отлаживать программу можно на массивах, заполненных случайными числами.)

2.5 Домашнее задание (ДЗ) - 11 Неделя

Задание для самостоятельной работы 15

Тема 15: Усовершенствованные методы сортировки.

1) На основе программы работы с массивом, которая описана задании для самостоятельной работы к тематическому занятию 14, составить функцию сортировки массива

методом, указанным в задании к лабораторной работе 15. (Функция должна работать с массивами как с параметрами.)

2) Для одного и того же массива из N элементов сравнить работу простого и усовершенствованного методов сортировки, реализованных на тематических занятиях 14 и 15, по количеству сравнений и присваиваний. (Оба метода должны проводить сортировку по одинаковому условию – либо по убыванию, либо по возрастанию.)

2.6 Тестирование (Т) - 11 Неделя

Тестовое задание по разделу V.

Вопрос 1

Определить, сколько раз будет выполнена операция присваивания при сортировке числового массива по возрастанию методом вставки:

```
int a[4] = { 4, 3, 2, 1 };
```

(Пояснение. Учитываются только те операции присваивания, в которых участвуют элементы массива. Если для обмена местами двух элементов массива используется дополнительная переменная, то необходимо учитывать каждое выполнение операции присваивания с ней.)

- 12
- 9
- 3
- 6
- 15
- 18

Вопрос 2

Определить, сколько раз будет выполнена операция присваивания при сортировке числового массива по возрастанию методом выбора:

```
int a[4] = { 3, 4, 2, 1 };
```

(Пояснение. Учитываются только те операции присваивания, в которых участвуют элементы массива. Если для обмена местами двух элементов массива используется дополнительная переменная, то необходимо учитывать каждое выполнение операции присваивания с ней.)

- 9
 - 12
 - 3
 - 6
 - 15
 - 18
-

Вопрос 3

Определить, сколько раз будет выполнена операция присваивания при сортировке числового массива по возрастанию методом «пузырька»:

```
int a[4] = { 4, 3, 2, 1 };
```

(Пояснение. Учитываются только те операции присваивания, в которых участвуют элементы массива. Если для обмена местами двух элементов массива используется дополнительная переменная, то необходимо учитывать каждое выполнение операции присваивания с ней.)

- 18
 - 9
 - 3
 - 6
 - 15
 - 12
-

Вопрос 4

Определить порядок сложности алгоритма, реализованного в форме следующего фрагмента кода:

```
i = 0;
while (i < n) {
    a[i][k] += i;
    i++;
}
for (j=0; j<n; ++j)
    a[k][j] *= j;
```

- $O(n)$
 - $2 \cdot O(n)$
 - $O(n^2)$
 - $O(1)$
 - $O(\log_2 n)$
 - $O(2^n)$
-

Вопрос 5

Определить, сколько раз будет выполнена перестановка двух элементов местами (swap) при сортировке числового массива по возрастанию методом быстрой сортировки Хоара (QuickSort):

```
int a[5] = { 5, 1, 3, 4, 2 };
```

(Пояснение. Учитываются только перестановки двух различных элементов массива.)

- 2
 - 3
 - 4
 - 5
 - 6
 - 7
-

Вопрос 6

Определить, сколько раз будет выполнена перестановка двух элементов местами (swap) до начала заполнения вспомогательного массива при сортировке числового массива по возрастанию методом сортировки слиянием (MergeSort):

```
int a[5] = { 5, 4, 1, 3, 2 };
```

(Пояснение. Учитываются только перестановки двух различных элементов массива.)

- 2
 - 1
 - 3
 - 4
 - 5
 - ни одного раза
-

Вопрос 7

Объявлены три массива:

```
int a[8] = { 1, 5, 2, 6, 8, 4, 3, 7 };
```

```
int b[8] = { 1, 2, 5, 4, 3, 8, 6, 7 };
```

```
int c[8] = { 1, 4, 2, 7, 5, 3, 8, 6 };
```

Указать массив(-ы), который(-ые) не образует пирамиду, строящуюся для проведения пирамидальной сортировки (HeapSort) по возрастанию.

- c
 - b
 - a
 - a и b
 - a и c
 - b и c
-

Вопрос 8

Объявлены три массива:

```
int a[8] = { 1, 4, 2, 7, 6, 3, 5, 8 };
```

```
int b[8] = { 1, 4, 2, 7, 5, 3, 8, 6 };
```

```
int c[8] = { 1, 2, 4, 5, 3, 8, 7, 6 };
```

Указать массив(-ы), который(-ые) не образует пирамиду, строящуюся для проведения пирамидальной сортировки (HeapSort) по возрастанию.

- b
 - a
 - c
 - a и b
 - a и c
 - b и c
-

Раздел 3 Многомерные массивы. Структуры. (Темат.занятия 16-18)

3.1 Контроль по итогам (КИ) - 15 Неделя

Текущий контроль успеваемости заключается в проверке выполнения заданий на лабораторных работах, проверке домашних заданий (для самостоятельной работы) с учетом своевременности их выполнения. В конце освоения раздела проводится тестирование.

Контроль освоения раздела дисциплины проводится преподавателем в форме контроля по итогам (КИ). Баллы за каждый раздел дисциплины вычисляются путем суммирования баллов, полученных студентом, за каждое входящее в данный раздел мероприятие текущего контроля: задания для лабораторных работ (ЛР), задания для самостоятельных работ (ДЗ), тестовое задание (Т).

Критерии оценки заданий для лабораторных работ (ЛР) и заданий для самостоятельных работ (ДЗ).

Программа, составленная студентом в ходе выполнения задания, должна соответствовать следующим требованиям:

- выполнены все условия, указанные в задании;
- компиляция и компоновка проходят успешно;
- во время выполнения программы не происходят сбои и зависания;
- интерфейс ввода-вывода соответствует требованиям для автоматической проверки решений;
- автоматическая проверка решений выполняется полностью и успешно;
- текст исходного кода программы оформлен в соответствии с требованиями структурного программирования;
- программа содержит комментарий к самым важным фрагментам кода.

За невыполнение каждого из перечисленных требований оценка решения может быть снижена на 1 балл. Штрафные (отрицательные) баллы не начисляются.

Критерии оценки тестовых заданий (Т).

Тестирование студентов в конце освоения каждого раздела дисциплины проводится на занятиях в компьютерном классе с помощью электронной обучающей системы.

Тестовое задание считается выполненным успешно, если студент набрал не менее 60% от максимального балла (округление результата проводится по обычным правилам).

Аттестация студента по разделу.

Контроль по итогам (КИ) считается пройденным успешно, а студент аттестованным по разделу, если он набрал не менее 60% от максимально возможного балла. Для данного раздела аттестационный балл равен **17**.

Оценка уровня сформированности каждой компетенции может быть получена путем вычисления процентной доли суммы набранных студентом баллов от максимально возможной суммы баллов за задания раздела, которые входят в перечень контрольных мероприятий соответствующей компетенции (округление проводится по обычным правилам).

3.2 Лабораторная работа (ЛР) - 12 Неделя

Задание для лабораторной работы 16

Тема 16: Массивы указателей.

Имеется два отсортированных исходных массива целых чисел: первый – по возрастанию (неубыванию), второй – по убыванию (невозрастанию). Составить программу, которая создает массив указателей на элементы обоих исходных массивов, упорядоченные по неубыванию. Т.е. эти указатели должны ссылаться на элементы исходных массивов так, чтобы при последовательном выводе массива указателей получилась общая последовательность всех элементов обоих исходных массивов, упорядоченная по неубыванию.

При этом запрещено: а) проводить перестановку элементов исходных массивов, б) объявлять и использовать дополнительные массивы. Количество элементов исходных массивов и их значения задает пользователь.

3.3 Домашнее задание (ДЗ) - 13 Неделя

Задание для самостоятельной работы 16

Тема 16: Массивы указателей.

1) Пользователь вводит слова, которые помещаются в массив символов одно за другим в порядке ввода, в конце каждого слова помещается нулевой символ ('\0'). Указатели на слова помещаются в массив указателей.

Составить программу, которая, используя массив указателей, проводит сортировку слов по алфавиту (при этом массив символов остается неизменным). Вывести отсортированный список слов на экран.

Реализовать то же самое, если слова не вводятся пользователем, а хранятся в текстовом файле, каждое в отдельной строке. При этом файл остается неизменным.

Все законченные действия должны быть реализованы в виде отдельных функций, а обмен данными между ними должен осуществляться через передачу параметров.

2) Пусть данные слова являются фамилиями участников конференции. Чтобы организовать массовое обслуживание участников одновременно в трех пунктах обслуживания, необходимо разделить их на две приблизительно равные части по фамилиям. Для этого необходимо разбить отсортированный список фамилий участников по первой букве (например: А–L, М–Z) так, чтобы разница между количеством участников в каждой из трех частей была наименьшей. При этом все участники, фамилии которых начинаются на одну и ту же букву, должны попасть в один пункт обслуживания.

Для каждой из трех частей списка вывести на экран интервал первых букв фамилий и количество участников (например, так: "А-L 35", "М-Z 37").

3.4 Лабораторная работа (ЛР) - 13 Неделя

Задание для лабораторной работы 17

Тема 17: Многомерные массивы. Матрицы.

Составить программу, которая создает матрицу фиксированной размерности $m \times n$ (значения m и n задаются пользователем).

1) Заполнить созданную матрицу целыми числами, введенными пользователем. Порядок заполнения элементов матрицы указан в таблице. Вывести матрицу на экран.

1	2	3	4
8	7	6	5
9	10	11	12
16	15	14	13

2) Определить номер строки, сумма элементов которой максимальна. Если таких строк несколько, то найти номера каждой из них.

(Отлаживать программу можно на матрицах, заполненных случайными числами.)

3.5 Домашнее задание (ДЗ) - 14 Неделя

Задание для самостоятельной работы 17

Тема 17: Многомерные массивы. Матрицы.

Составить программу для работы с квадратной матрицей фиксированной размерности ($n \times n$), которая содержит функции, позволяющие: 1) заполнять матрицу случайными числами из заданного интервала; 2) выводить матрицу на экран (построчно); 3) сохранять и считывать матрицу из текстового файла. А также составить функции, которые позволяют выполнять матричные операции: 4) транспонирование; 5) сложение; 6) умножение. Все функции должны работать с матрицами как с параметрами.

С помощью разработанных функций для заданной матрицы A найти значение выражения $A \times A^T + A = ?$

3.6 Лабораторная работа (ЛР) - 15 Неделя

Задание для лабораторной работы 18

Тема 18: Структуры.

Сведения о студентах содержат следующую информацию: фамилия, имя, курс, оценки по предметам: математика, физика, информатика (в баллах, от 0 до 100). Создать массив структур (из 8-12 элементов) и заполнить его данными.

Найти фамилию и имя студента 1-го курса, который учится лучше остальных.

3.7 Домашнее задание (ДЗ) - 15 Неделя

Задание для самостоятельной работы 18

Тема 18: Структуры.

Сведения об участниках конкурса содержат следующую информацию: фамилия, имя, адрес (город, улица, номер дома, номер квартиры), секция, набранный балл. Составить программу, которая считывает эти данные из текстового файла в массив структур (не менее 20-и элементов). Составить функции, которые позволяют:

- 1) найти средний балл по каждой секции;
- 2) определить фамилии победителей по каждой секции;

3) определить город(-а), в котором проживает наибольшее количество победителей.

3.8 Тестирование (Т) - 15 Неделя

Тестовое задание по разделу VI.

Вопрос 1

Объявлен массив

```
char a[2][3] = { 1, 2, 3, 4, 5, 6 };
```

Определить, какой из указателей содержит адрес элемента 5 .

- `*(a+1)+1`
- `*(a[1]+1)`
- `*a[1]+1`
- `*(a+2)+1`
- `*(a[2]+1)`
- `*a[2]+1`

Вопрос 2

Объявлен массив

```
char a[2][3] = { 10, 20, 30, 40, 50, 60 };
```

Определить, какое из выражений возвращает значение, равное 50 .

- `*(*(a+1)+1)`
- `*(a+1)+1`
- `*a[1]+1`
- `*a[1][1]`
- `*(a+1)+1`
- `*(a[1]+1)+1`

Вопрос 3

В программе объявлены два массива:

```
int a[5] = { 4, 3, 5, 2, 1 };
```

`arrptr` – массив из 5-и указателей на переменные типа `int`.

Массив `arrptr` заполнен указателями на элементы массива `a` в порядке возрастания значений элементов массива `a`.

Выбрать выражение, которое возвращает значение, тождественное адресу максимального элемента массива `a`.

- `*(arrptr+4)`
 - `*arrptr`
 - `*(arrptr+1)`
 - `*(arrptr+2)`
 - `*(arrptr+3)`
 - `*(arrptr+5)`
-

Вопрос 4

В программе объявлен массив:

```
int m[2][3] = { 4, 6, 2, 3, 1, 5 };
```

Выбрать выражение, которое возвращает адрес минимального элемента массива `m`.

- `*m+4`
 - `*m+1`
 - `*m+2`
 - `*m+3`
 - `*m+5`
 - `*m`
-

Вопрос 5

В программе сделаны следующие объявления:

```
int a[10];  
int *arrptr[10];
```

Массив `a` заполнен целыми числами, а массив `arrptr` заполнен указателями на элементы массива `a`.

Указать выражение(-я), у которого(-ых) тип возвращаемого результата не является типом «указатель на `int`».

```
arrptr+5    /* 1 */  
arrptr[4]   /* 2 */  
*arrptr+3   /* 3 */
```

- только 1
 - только 2
 - только 3
 - 1 и 2
 - 1 и 3
 - 2 и 3
-

Вопрос 6

В программе объявлен массив:

```
int m[3][5];
```

Указать, какое из следующих выражений возвращает результат, равный результату выражения `m[1]`.

- `*m+5`
 - `*m+3`
 - `*(m+2)+1`
 - `*(m+1)+2`
 - `*(m+3)+2`
 - `*(m+2)+3`
-

Вопрос 7

В программе сделаны следующие объявления

```
struct tStaff {          /* Персонал */
    char Head[30];        /* руководитель */
    char Deputy[30];      /* зам.руководителя */
    char Accountant[30];  /* гл.бухгалтер */
    int Number;           /* кол-во сотрудников */
};
struct tCompany {        /* Компания */
    char Name[100];       /* название */
    int RegNum;           /* рег.номер */
    char RegAddress[150]; /* юридич.адрес */
    struct tStaff Staff;  /* персонал */
} Consortium[5];         /* Консорциум из 5-и компаний */
```

Выбрать выражение, определяющее на какую букву начинается имя руководителя 2-й компании консорциума.

- Consortium[1].Staff.Head[0]
 - Consortium.Staff.Head[1][0]
 - Consortium.Staff.Head[0][1]
 - Consortium.Staff[1].Head[0]
 - Consortium[0].Staff.Head[1]
 - Consortium.Staff[0].Head[1]
-

Вопрос 8

В программе сделаны следующие объявления

```
struct tCompany {        /* Компания */
    char Name[100];       /* название */
    int RegNum;           /* рег.номер */
    char RegAddress[150]; /* юридич.адрес */
} Consortium[5];         /* Консорциум из 5-и компаний */
struct tCompany *pCompany;
```

Выбрать выражение, определяющее регистрационный номер 1-й компании консорциума, после выполнения оператора присваивания

```
pCompany = &Consortium[1];
```

- (*(pCompany-1)).RegNum
 - (*pCompany).RegNum
 - (*(pCompany+1)).RegNum
 - *(pCompany-1).RegNum
 - *pCompany.RegNum
 - *(pCompany+1).RegNum
-

1 Семестр

Зачет

Задания содержат задачу, для решения которой студенту необходимо составить, откомпилировать и выполнить программу на компьютере с использованием компилятора языка C (стандарт ANSI) и интегрированной среды разработчика (IDE). Время выполнения задания – два академических часа. Составленную студентом программу проверяет преподаватель.

Примеры содержания заданий.

Задание. Вариант 1

Из целочисленного массива A сформировать массив B таким образом, что каждый его элемент $B[i]$ является средним арифметическим всех элементов массива A, кроме i-го.

Задание. Вариант 2

Составить функцию, которая переводит переданное в нее число в двоичную систему счисления.

Задание. Вариант 3

Определить сумму всех трехзначных чисел, которые без остатка делятся на 3.

Критерии оценки экзаменационных заданий.

Программа, составленная студентом в ходе выполнения задания, должна соответствовать следующим требованиям:

- выполнены все условия, указанные в задании;
- компиляция и компоновка проходят успешно;
- во время выполнения программы не происходят сбои и зависания;
- интерфейс ввода-вывода соответствует требованиям для автоматической проверки решений;
- автоматическая проверка решений выполняется полностью и успешно;
- текст исходного кода программы оформлен в соответствии с требованиями структурного программирования;
- программа содержит комментарий к самым важным фрагментам кода.

За невыполнение каждого из перечисленных требований оценка решения может быть снижена на 1-2 балла.

Аттестация студента.

Задание считается выполненным успешно, а студент аттестованным, если он набрал не менее 60% от максимально возможного балла. В 1-м семестре аттестационный балл равен **12**.

2 Семестр

Экзамен

Во 2-м семестре студенты сдают **экзамен** по дисциплине. Экзамен проводится в компьютерном классе. Экзаменационные задания содержат задачу, для решения которой студенту необходимо составить, откомпилировать и выполнить программу на компьютере с использованием компилятора языка C (стандарт ANSI) и интегрированной среды разработчика (IDE). Время выполнения задания – два академических часа. Составленную студентом программу проверяет преподаватель.

Примеры содержания экзаменационных заданий.

Экзаменационное задание. Вариант 1

Вычислить приближенное значение бесконечной суммы:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

Экзаменационное задание. Вариант 2

Для матрицы, заполненной случайными числами, определить номер строки, сумма элементов которой максимальна, и поменять ее местами с первой строкой.

Экзаменационное задание. Вариант 3

Для матрицы $n \times n$, заполненной случайными числами, разделить все элементы каждой строки на значение элемента данной строки, принадлежащего главной диагонали.

Критерии оценки экзаменационных заданий.

Программа, составленная студентом в ходе выполнения задания, должна соответствовать следующим требованиям:

- выполнены все условия, указанные в задании;
- компиляция и компоновка проходят успешно;
- во время выполнения программы не происходят сбои и зависания;
- интерфейс ввода-вывода соответствует требованиям для автоматической проверки решений;
- автоматическая проверка решений выполняется полностью и успешно;
- текст исходного кода программы оформлен в соответствии с требованиями структурного программирования;
- программа содержит комментарий к самым важным фрагментам кода.

За невыполнение каждого из перечисленных требований оценка решения может быть снижена на 1-2 балла.

Аттестация студента.

Экзаменационное задание считается выполненным успешно, а студент аттестованным, если он набрал не менее 60% от максимально возможного балла. Для экзамена во 2-м семестре аттестационный балл равен **12**.

Оценка уровня сформированности каждой компетенции может быть получена путем вычисления процентной доли суммы набранных студентом баллов от максимально возможной суммы баллов за задания семестра, которые входят в перечень контрольных мероприятий соответствующей компетенции (округление проводится по обычным правилам).