

**Перечень теоретических и практических заданий к экзамену
по ОП.04 Основы алгоритмизации и программирования
(2 курс, 4 семестр 2025-2026 уч. г.)**

Форма контроля: Практическая работа (Опрос)

Описательная часть: По выбору выполнить 1 теоретическое задание и 1 практическое задание

Перечень теоретических заданий:

Задание №1

Ответить на вопросы:

Вопрос 1 Для решения любой задачи с помощью компьютера необходимо выполнить следующие этапы:

Установите правильную последовательность этапов.

- | | |
|---|-------------------------------|
| 1 | принятие решений; |
| 2 | программирование; |
| 3 | математическое моделирование; |
| 4 | алгоритмизация задач; |
| 5 | постановка задачи; |
| 6 | анализ результатов. |

Вопрос 2 Соотнесите свойства алгоритма с их описанием:

- | | |
|------------------|--|
| Результативность | - алгоритм должен приводить к решению задачи обязательно за конечное время; |
| Конечность | - неоднозначность толкования алгоритма недопустима; |
| Эффективность | - алгоритм должен обеспечить выдачу результата решения задачи на печать, на экран монитора или в файл; |
| Массовость | - правильный результат по алгоритму получен для одних исходных данных, то правильный результат по этому же алгоритму должен быть получен и для других исходных |

Определенность

данных, допустимых в данной задаче;
- позволяет решить задачу за приемлемое для разработчика время;

Вопрос 3 Выберите тип алгоритма, описанного ниже:

Набор команд (указаний), выполняемых последовательно друг за другом

1. линейный;
2. разветвляющийся;
3. циклический.

Вопрос 4 Выберите тип алгоритма, описанного ниже:

Алгоритм, содержащий хотя бы одну проверку условия, в результате которой обеспечивается переход на один из возможных вариантов решения

1. линейный;
2. разветвляющийся;
3. циклический.

Вопрос 5 Выберите тип алгоритма, описанного ниже:

Алгоритм, предусматривающий многократное повторение одного и того же действия над новыми исходными данными

1. линейный;
2. разветвляющийся;
3. циклический.

Оценка	Показатели оценки
3	Даны правильные ответы на 3 вопроса
4	Даны правильные ответы на 4 вопроса
5	Даны правильные ответы на 5 вопросов

Задание №2

Ответить на вопросы:

1. Объясните на примере принцип работы полной и краткой формы условного оператора.
2. как показать условный оператор на схеме алгоритма? Приведите примеры.
3. Можно ли использовать условный оператор внутри другого условного оператора (например, в ветви else)?

Оценка	Показатели оценки
3	Даны ответы на 2 вопроса, примеры не приведены
4	Даны ответы на все вопросы, примеры не приведены
5	Даны ответы на все вопросы, приведены примеры

Задание №3

Ответить на вопросы:

1. Из чего состоит итерация для оператора while? Объясните на примере.
2. Объясните механизм работы оператора do-while на примерах.
3. Запишите в общем виде оператор цикла for. Какие ключевые слова при этом используются.

Оценка	Показатели оценки
3	Даны ответы на 2 вопроса, примеры не приведены
4	Даны ответы на все вопросы, примеры не приведены
5	Даны ответы на все вопросы, приведены примеры

Задание №4

Ответить на вопросы:

Вопрос 1 Установите правильную хронологию создания языков программирования:

1	C#;
2	бейсик (Basic);
3	паскаль (Pascal);
4	C++;

Вопрос 2 Соотнесите годы создания языков программирования

1963	C#;
------	-----

1971	бейсик (Basic);
1984	паскаль (Pascal);
2000	C++.

Вопрос 3 Выберите процедурный язык программирования:

1. C++;
2. Basic;
3. Java.

Вопрос 4 Какие языки программирования предназначены для решения задач искусственного интеллекта:

1. Commonlisp.
2. Planner.
3. Occam.
4. C++.
5. Java.

Вопрос 5 Какие языки программирования предназначены для разработки программ-оболочек, разработки систем:

1. Commonlisp.
2. Planner.
3. Occam.
4. C++.
5. Java.

Оценка	Показатели оценки
3	Даны правильные ответы на 3 вопроса
4	Даны правильные ответы на 4 вопроса
5	Даны правильные ответы на 5 вопросов

Задание №5

Ответить на вопросы:

Вопрос 1. Выберите видовые отличия термина "Язык программирования"

1. предназначен для записи программ;

2. имеет определенный синтаксис;
3. предназначен для передачи информации;
4. машиноориентированный, низкого уровня.

Вопрос 2 Выберите видовые отличия термина "Алгоритм"

1. ориентирована на пользователя;
2. выполняется за конечное число шагов;
3. предназначен для записи программ;
4. имеет определенный синтаксис.

Вопрос 3. Выберите правильное определение для термина "Математическая модель"

1. модель, описывающая математическими формулами соотношения между количественными характеристиками объекта моделирования
2. объект, который воспроизводит наиболее характерные свойства реального процесса или устройства или концепции
3. модель, которая определяет уровни взаимодействия систем и их стандартные названия и функции

Вопрос 4 Выберите родовое понятие для термина "Моделирование"

1. система познания;
2. набор объектов;
3. множество методов.

Вопрос 5 Соотнесите термины и родовые понятия

алфавит		множество
идентификатор		набор символов
константа		величина
ключевые слова		единица языка

Вопрос 6 Соотнесите термины и видовые отличия

--	--	--

идентификатор		является именем объектов программы
константа		имеет постоянное значение
ключевые слова		имеют специальное значение для компилятора

Вопрос 7 Выберите правильный вариант использования условного оператора if для нахождения $\text{MAX}\{C*D, E+F\}$:

1. `if (C*D>E+F) Max=C*D; else Max:=E+F;`
2. `X=C*D; Y=E+F; if (X>Y) Max=X; else MAX=Y;`
3. оба варианта правильные.

Вопрос 8 Выберите пример правильного идентификатора в языке C++

1. `Fr_5;`
2. `10Sd;`
3. `scanf.`

Вопрос 9 Какие значения примут переменные S, N и P после выполнения следующего фрагмента программы,

если в переменную Y поочередно ввести следующие значения: -2; 0; -5; 6; 0?

...

`S=0; N=0; P=1;`

`for (i=1; i<=5; i++)`

`{`

`scanf ("%d", &Y);`

`if (Y>0)`

`S=S+Y;`

`else if (Y=0)`

N=N+1;

else P=P*Y;

}

...

1. S=-7; N=1; P=0;
2. S=6; N=2; P=10;
3. S=-1; N=4; P=10.

Вопрос 10 Что делает следующий фрагмент программы с массивом A, содержащим N элементов?

...

```
for (i=0; i<N/2; i++)
```

```
{ M=A[i];
```

```
A[i]=A[N-1-i];
```

```
A[N-1-i]=M;
```

```
}
```

...

1. присваивает каждому элементу массива значение следующего элемента, а последнему - значение 1-го элемента;
2. записывает элементы массива в обратном порядке;
3. фрагмент содержит ошибки и работать не будет.

Вопрос 11 Укажите правильный вариант определения класса в программе:

1. class Test

```
{  
public:  
int a;  
};
```

2. class Test

```
{  
private:  
float x;  
}
```

3. оба варианта правильные.

Оценка	Показатели оценки
3	Даны правильные ответы на 5-7 вопросов
4	Даны правильные ответы на 8-10 вопросов
5	Даны правильные ответы на 11 вопросов

Задание №6

Ответить на вопросы:

1. Что такое зарезервированные (ключевые слова)? Приведите примеры ключевых слов.
2. Верно ли, что в любой программе на C++ обязательно должны использоваться переменные и константы? Почему?
3. Чем отличаются операции "/" и "%"? Приведите примеры.

Оценка	Показатели оценки
3	Даны ответы на 2 вопроса, примеры не приведены
4	Даны ответы на все вопросы, примеры не приведены
5	Даны ответы на все вопросы, приведены примеры

Задание №7

Ответить на вопросы:

1. Данные каких типов могут быть элементами массива? Приведите примеры.
2. Как получить доступ к конкретному элементу массива? Приведите примеры.
3. Можно ли в программе объявить пользовательский тип "массива"? Как это сделать? Приведите примеры.

Оценка	Показатели оценки
3	Даны ответы на 2 вопроса, примеры не приведены
4	Даны ответы на все вопросы, примеры не приведены
5	Даны ответы на все вопросы, приведены примеры

Задание №8

Ответить на вопросы:

Вопрос 1. В чем разница между формальными и фактическими параметрами?

1. никакой разницы нет;
2. формальные параметры используются при описании подпрограммы, а фактические — при вызове подпрограммы;
3. фактические параметры используются при описании подпрограммы, а формальные — при вызове подпрограммы.

Вопрос 2. В чем разница между глобальными и локальными переменными?

1. разницы нет;
2. глобальные переменные не могут использоваться в подпрограммах, для этого служат локальные переменные;
3. глобальные переменные могут использоваться во всех подпрограммах и в функции `main ()`, а локальные переменные только в своей подпрограмме.

Вопрос 3. Выберите определение функции, не содержащее ошибок:

1. `int Area(int A, int B)`
`{ float S;`
`Area=A*B;}`

2. `int Area(int A, int B)`
`{ int S;`
`S=A*B; return S;}`

3. `int Area(int A, int B)`
`{ int S;`
`S=A*B;`
`return Area;`

Вопрос 4. Что будет выведено на экран в результате работы следующей программы?

```
int A;  
  
void Prim(int A)  
{A=5;  
  
printf(" %d",A);  
  
}
```

```

int main()

(int A=10;

Prim(A);

printf ( " %d", A) ; return 0;

}

```

1. 10 10;
2. 5 10;
3. 5 5.

Вопрос 5. Должны ли имена параметров, указанных в прототипе, определении и вызове функции, соответствовать друг другу?

1. да;
2. нет;
3. должны соответствовать в прототипе и определении.

Оценка	Показатели оценки
3	Даны правильные ответы на 3 вопроса
4	Даны правильные ответы на 4 вопроса
5	Даны правильные ответы на 5 вопросов

Задание №9

Ответить на вопросы:

1. Что такое возвращаемое значение в функции? Приведите примеры.
2. Напишите структуру функции в общем виде и объясните, для чего используется каждый раздел функции. Приведите примеры определения функции.
3. Что такое формальные и фактические параметры в функции? Приведите примеры.

Оценка	Показатели оценки
3	Даны ответы на 2 вопроса, примеры не приведены
4	Даны ответы на все вопросы, примеры не приведены
5	Даны ответы на все вопросы, приведены примеры

Задание №10

Ответить на вопросы:

1. Что такое "структура" в программировании с точки зрения типов данных?
2. Какие требования предъявляются к именам полей структур? Поясните на примерах.
3. Как правильно присваивать значения полям структуры? Поясните на примерах.

Оценка	Показатели оценки
3	Даны ответы на 2 вопроса, примеры не приведены
4	Даны ответы на все вопросы, примеры не приведены
5	Даны ответы на все вопросы, приведены примеры

Задание №11

Ответить на вопросы:

Вопрос 1. Принцип инкапсуляции обеспечивает:

1. объединение данных и методов работы с ними в классе;
2. доступ к членам класса;
3. сокрытие данных внутри класса.

Вопрос 2. Укажите правильный вариант инициализации целочисленных констант А и В в классе:

1. `klass::klass(int a, int b) : A(a), B(b)`
{
...
}

2. `klass::klass(int a, int b) : A= a, B = b`
{
...
}

3. `klass::klass(int a, int b) : A(a) B(b)`
{
...
}

Вопрос 3. Наследование — это:

1. возможность использования базовых библиотек языка C++ в своих программах;
2. условия, описывающие последовательность вызова конструкторов для объектов классов, используемых в программе;
3. механизм создания производных классов, на базе уже имеющихся.

Вопрос 4. Укажите верную последовательность выполнения деструкторов:

1. сначала выполняются операторы деструктора базового класса, затем операторы деструктора порожденного класса;
2. сначала выполняются операторы деструктора порожденного класса, затем операторы деструктора базового класса;
3. операторы деструктора порожденного класса выполняются одновременно с операторами деструктора базового класса.

Вопрос 5. Полиморфизм — это:

1. возможность программного кода работать с разными объектами одинаковым образом;
2. возможность изменения программного кода в зависимости от решаемых задач;
3. возможность доработки программного кода в случае необходимости.

Оценка	Показатели оценки
3	Даны правильные ответы на 3 вопроса
4	Даны правильные ответы на 4 вопроса
5	Даны правильные ответы на 5 вопросов

Перечень практических заданий:

Задание №1

Разработайте алгоритм для задачи:

Определить, принадлежит ли точка $A\{x, y\}$ заданной фигуре.

Оценка	Показатели оценки
3	Алгоритм составлен с двумя ошибками
4	Алгоритм составлен с одной ошибкой

5	Алгоритм составлен без ошибок
---	-------------------------------

Задание №2

Разработайте алгоритм для задачи:

Ввести два числа и символ — знак арифметической операции. В зависимости от введенного знака операции вычислить значение арифметического выражения.

Оценка	Показатели оценки
3	Алгоритм составлен с двумя ошибками
4	Алгоритм составлен с одной ошибкой
5	Алгоритм составлен без ошибок

Задание №3

Разработайте алгоритм для задачи:

Даны три числа: a , b , c . Определить, могут ли они быть сторонами треугольника, и если могут, то определить его тип: равносторонний, равнобедренный, разносторонний. (Условие существования треугольника: сумма длин любых двух сторон треугольника превышает длину 3-й стороны. Следует также учесть случай, когда длина одной из сторон равна нулю или имеет отрицательное значение.)

Оценка	Показатели оценки
3	Алгоритм составлен с двумя ошибками
4	Алгоритм составлен с одной ошибкой
5	Алгоритм составлен без ошибок

Задание №4

Разработайте алгоритм для задачи:

Ввести два целых числа. Вывести в порядке убывания все числа, лежащие между ними, и количество этих чисел. Каждое третье число не печатать и не учитывать.

Оценка	Показатели оценки
--------	-------------------

3	Алгоритм составлен с двумя ошибками
4	Алгоритм составлен с одной ошибкой
5	Алгоритм составлен без ошибок

Задание №5

Разработайте алгоритм для задачи:

В 1202 г. итальянский математик Леонард Фибоначчи подсчитывал, на сколько увеличивается число кроликов в хозяйстве каждый год. При этом он получил последовательность такого вида: 1, 1, 2, 3, 5, 8, 13, 21, 34 Написать программу, которая для заданного числа A выводит N членов последовательности Фибоначчи.

Оценка	Показатели оценки
3	Алгоритм составлен с двумя ошибками
4	Алгоритм составлен с одной ошибкой
5	Алгоритм составлен без ошибок

Задание №6

Разработайте алгоритм для одной из задачи с использованием программы для графического отображения алгоритмов:

Определить, принадлежит ли точка $A\{x, y\}$ заданной фигуре.

Оценка	Показатели оценки
3	Алгоритм составлен с двумя ошибками
4	Алгоритм составлен с одной ошибкой
5	Алгоритм составлен без ошибок

Задание №7

Разработайте алгоритм для одной из задачи с использованием программы для графического отображения алгоритмов:

Ввести два числа и символ — знак арифметической операции. В зависимости от введенного знака операции вычислить значение арифметического выражения

Оценка	Показатели оценки
3	Алгоритм составлен с двумя ошибками
4	Алгоритм составлен с одной ошибкой
5	Алгоритм составлен без ошибок

Задание №8

Разработайте алгоритм для одной из задачи с использованием программы для графического отображения алгоритмов:

Даны три числа: a , b , c . Определить, могут ли они быть сторонами треугольника, и если могут, то определить его тип: равносторонний, равнобедренный, разносторонний. (Условие существования треугольника: сумма длин любых двух сторон треугольника превышает длину 3-й стороны. Следует также учесть случай, когда длина одной из сторон равна нулю или имеет отрицательное значение.)

Оценка	Показатели оценки
3	Алгоритм составлен с двумя ошибками
4	Алгоритм составлен с одной ошибкой
5	Алгоритм составлен без ошибок

Задание №9

Разработайте алгоритм для одной из задачи с использованием программы для графического отображения алгоритмов:

Ввести два целых числа. Вывести в порядке убывания все числа, лежащие между ними, и количество этих чисел. Каждое третье число не печатать и не учитывать

Оценка	Показатели оценки
3	Алгоритм составлен с двумя ошибками
4	Алгоритм составлен с одной ошибкой
5	Алгоритм составлен без ошибок

Задание №10

Разработайте алгоритм для одной из задачи с использованием программы для графического отображения алгоритмов:

В 1202 г. итальянский математик Леонард Фибоначчи подсчитывал, на сколько увеличивается число кроликов в хозяйстве каждый год. При этом он получил последовательность такого вида: 1, 1, 2, 3, 5, 8, 13, 21, 34 Написать программу, которая для заданного числа A выводит N членов последовательности Фибоначчи.

Оценка	Показатели оценки
3	Алгоритм составлен с двумя ошибками
4	Алгоритм составлен без ошибок
5	Алгоритм составлен без ошибок

Задание №11

Определить функцию сложности алгоритма по результатам эксперимента:

N	Количество перестановок
5	62

Оценка	Показатели оценки
3	Задача решена с ошибками
4	Задача решена, есть замечания по расчетам
5	Задача решена верно

Задание №12

Определить функцию сложности алгоритма по результатам эксперимента:

N	Время работы, с
1000	0,134

Оценка	Показатели оценки

3	Задача решена с ошибками
4	Задача решена, есть замечания по расчетам
5	Задача решена верно

Задание №13

Напишите инструкции по работе со средой программирования Visual Studio

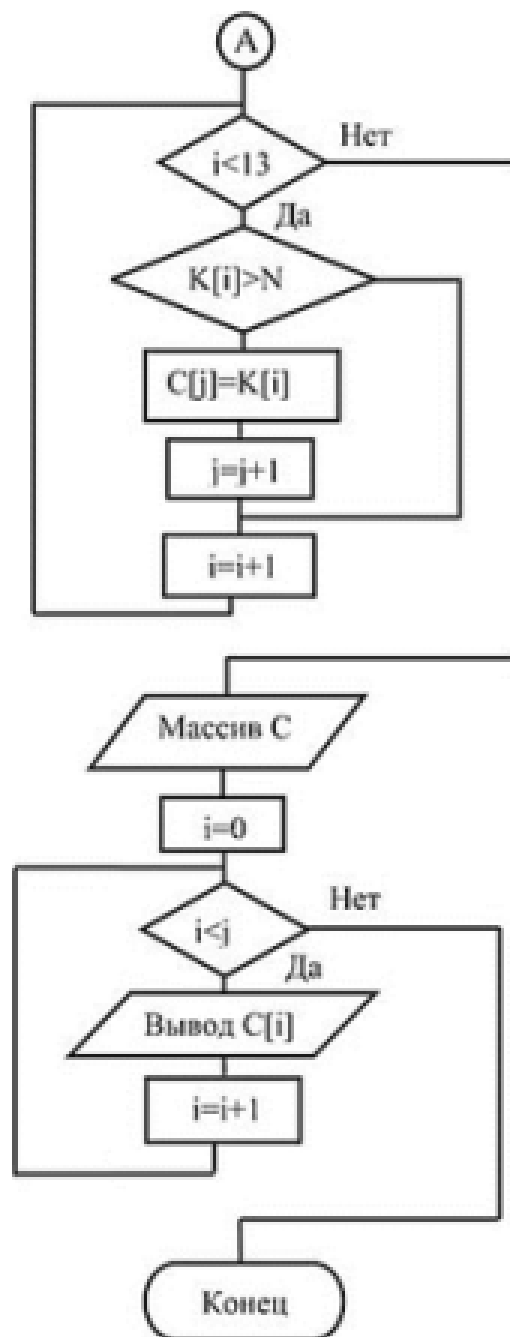
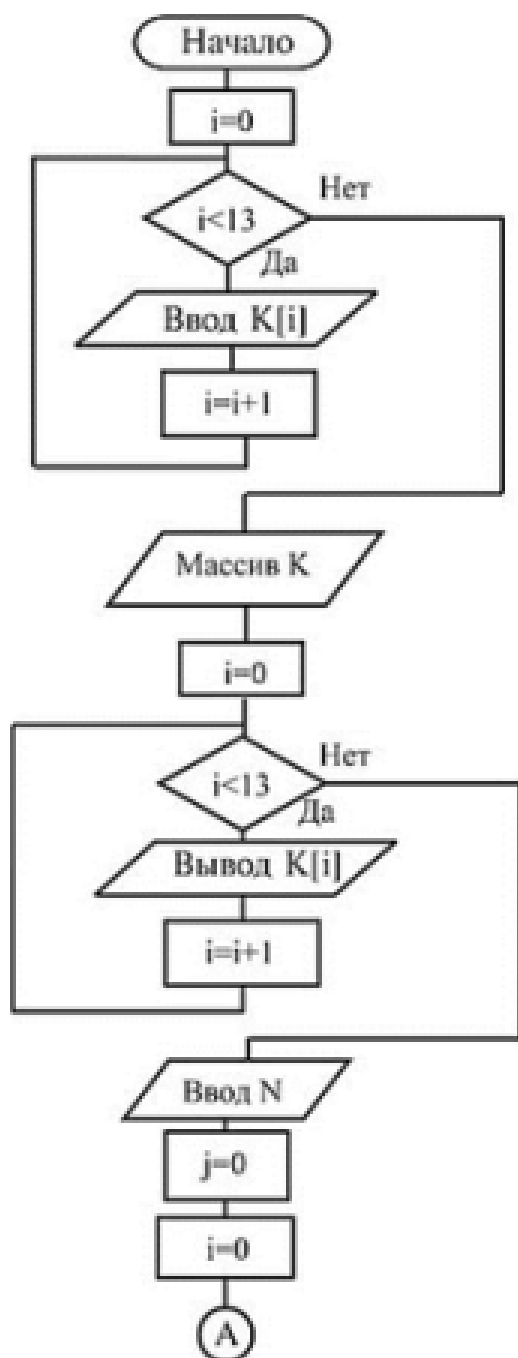
- добавление файлов в созданный проект;
- выполнение отладки программы;
- выполнение программы по шагам.

Оценка	Показатели оценки
3	Написана одна из инструкций.
4	Написано две инструкции.
5	Написаны все инструкции.

Задание №14

Написать программу для задачи, по приведенной схеме алгоритма:

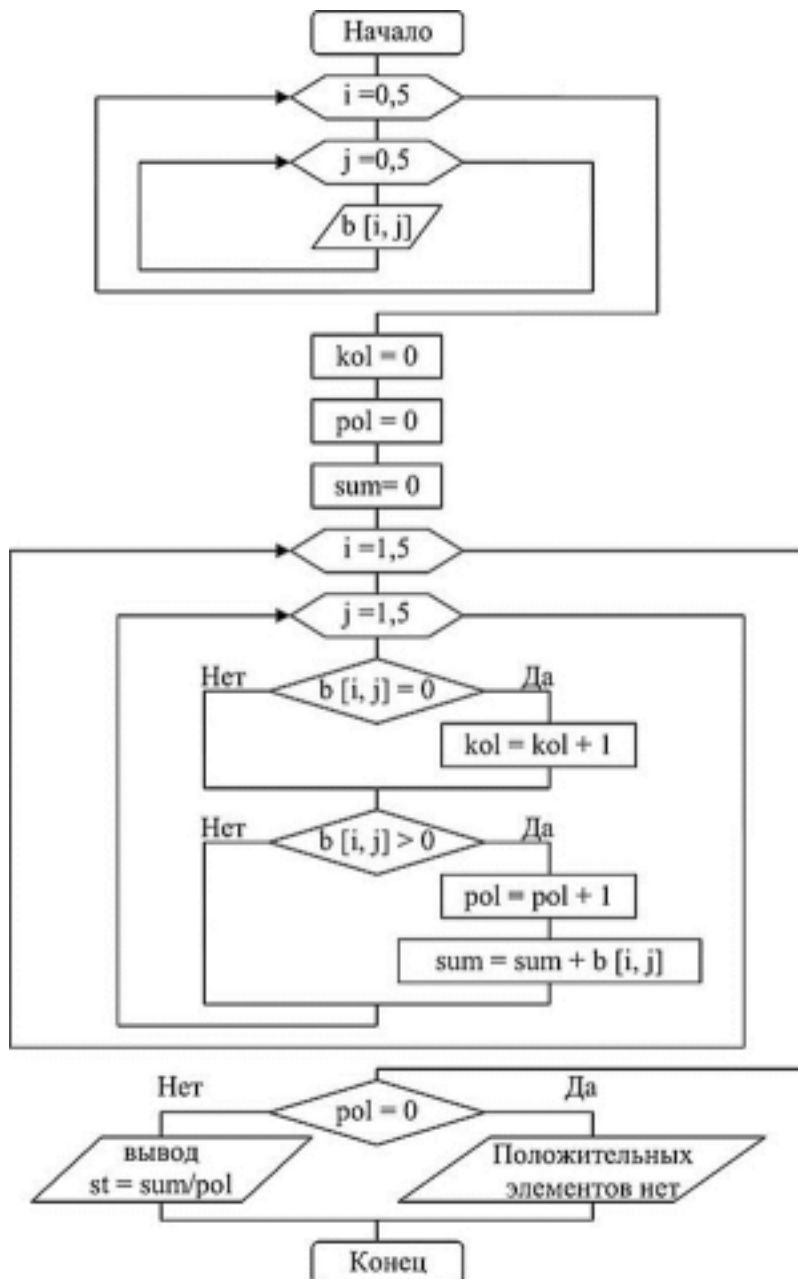
Дан массив K из 13 элементов и целое число N (вводится с клавиатуры). Из элементов массива K , больших числа N , сформировать массив C . Вывести оба массива на экран.



Оценка	Показатели оценки
3	для представленного алгоритма программа написана, допущены ошибки
4	для представленного алгоритма программа написана, не предусмотрен один из критериев
5	для представленного алгоритма программа написана, учтены все из вариантов решения

Написать программу для задачи, по приведенной схеме алгоритма:

Найти сумму положительных элементов двумерного массива.



Оценка	Показатели оценки
3	для представленного алгоритма программа написана, допущены ошибки
4	для представленного алгоритма программа написана, не предусмотрен один из критериев
5	для представленного алгоритма программа написана, учтены все из вариантов решения

Задание №16

Оформите представленный код программы в соответствии со стандартом кодирования:

Вершины треугольника заданы массивом точек. Точки являются объектом класса CPoint. Написать функцию, вычисляющую площадь треугольника по формуле Герона:

```
#include #include class CPoint

{ private: int x, y; public:

CPoint(int _x, int _y)

(x = _x; y = _J/;}

CPoint()

{x = 0; y = 0; } void SetX(int _x)

{x - _x;}

void SetY(int _y)

(Y = _Y; > int GetX()

{ return x; } int GetY() { return y; } };

float length(CPoint pi, CPoint p2)

{

int x1, x2, y1, y2, L; x1 = pi.GetX(); y1 = pi.GetY(); x2 = p2.GetX(); y2 = p2.GetY();

L = sqrt(pow(x2-x1, 2.0) + pow(y2-y1, 2.0)); return L;

float square(CPoint *mas)

{float L1, L2, L3, p;

L1 = length(mas[0], mas[1]);

L2 = length(mas[1], mas[2]);

L3 = length(mas[2], mas[0]); p = (L1 + L2 + L3)/2.0; if (p*(p-L1)*(p-L2)*(p-L3)>=0)
```

```

return sqrt(p*(p - L1)*(p - L2)*(p - L3)); else (printf("Треугольник построить нельзя! "); return -1;

}} int main(){

CPoint mas [3]; int x, y;

for (int i = 0; i<3; i++){printf("First point:"); scanf("%d%d", &x, &y); mas [i] .SetX(x); mas[i].SetY(y);}

printf("S = %3.2f ", square(mas)); return 0;}

```

Оценка	Показатели оценки
3	программа оформлена частично в соответствии со стандартом кодирования
4	программа оформлена в соответствии со стандартом кодирования, с небольшими недочетами
5	программа оформлена полностью в соответствии со стандартом кодирования

Задание №17

Выполните проверку и отладку следующего кода программы

Написать класс «герой». Члены класса: имя героя, его возраст. Написать класс «параметры». Члены класса: сила, защита, скорость. На базе этих двух классов написать класс «воин», включающий в себя все перечисленные члены.

Иерархия классов будет выглядеть следующим образом:

```
#include
```

```
#include
```

```
class Hero //начало определения 1-го базового класса Него
```

```
{
```

```
protected;
```

```
char name [32 ] ; //защищенная член-переменная name класса Него
```

```
int age; //защищенная член-переменная аде класса Него
```

```
public:
```

```

Hero () //определение конструктора класса Hero без параметров

{

strcpy(name, "NoName");

//копирование в член-переменную name текста NoName age = 0; //присвоить члену-переменной age
значения 0

}

Hero(char* name, int age)

//определение конструктора класса Hero с двумя параметрами

{

strcpy (this ->name, name) ; /*копирование члена-переменной
name (this->name,) значения параметра name V

this->age = age;

//присвоить члену-переменной age (this->age^ значение параметра age

}

void info () //определения функции-члена info

{

printf("Hero:%s, %d ", name, age);

//вывод на экран названия класса и значений членов-переменных name и age

}

}; //конец определения базового класса Hero

class Parametry //начало определения базового класса Parametry

{

protected:

int si la; //защищенная член-переменная si la класса Parametry

int zashita;

```

```

//защищенная член-переменная zashita класса Parametry

int skorost;

//защищенная член-переменная skorost класса Parametr

public:

Parametry(int _sila = 0, int _zashita = 0, int _skoros = 0)

//конструктор класса Parametry со значениями поумолчаниюю

{

sila = _sila; zashita = _zashita; skorost = _skorost;

}

void info() //определение члена-функции info класса Parametry

printf("Parametry: %d, %d, %d ", sila, zashita,

skorost) ; /*вывод на экран названия класса и значений

членов-переменных sila, zashita, skorost*/

}

}; //конец определения класса Parametry

class Warrior : public Hero, Parametry

//начало определ. класса Warrior наследника классов Hero, Parametry { public:

/*определение конструктора класса Warrior инициализацией «родительских» конструкторов

классов Hero и Parametry*/

Warrior(char* _name, int _age, int _sila, int _zashita, int _skorost): Hero(_name, _age), Parametry(_sil,

_zashita, _skoros)

{ }

void info() //определение члена-функции info класса Warrior

{

printf("Warrior: ");

```

```

//вывод на экран названия класса Warrior Hero : : info () ; //вызов функции-члена info класса Hero
Parametry::info();

//вызов функции-члена info класса Parametry

}

}; //конец определения класса Warrior

int main()

Warrior w("Ivan", 30, 100, 300, 15); w.info(); return 0;

}

```

Оценка	Показатели оценки
3	проверка и отладка выполнена, программа работает с ошибками
4	проверка и отладка выполнена, программа работает с небольшими недочетами
5	проверка и отладка выполнена, программа работает верно