

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ
Кафедра веб-технологий и компьютерного моделирования**

А. С. Кравчук, А. И. Кравчук, Е. В. Кремень

Тест
**Введение в методы
программирования**

**Учебные материалы
для студентов специальности 6-05-0533-07
«Математика и компьютерные науки
(по профилизациям)»**

**МИНСК
2023**

УДК 004.42(075.8)
ББК 32.973-018я73-1
К78

Рекомендовано советом
механико-математического факультета БГУ
30 мая 2023 г., протокол № 9

Рецензент
профессор кафедры экономической информатики
Белорусского государственного экономического университета
кандидат технических наук, доцент *А. М. Седун*

Кравчук, А. С.
К78 Тест. Введение в методы программирования : учеб. материалы
для студентов специальности 6-05-0533-07 «Математика и
компьютерные науки (по профилизациям)» / А. С. Кравчук,
А. И. Кравчук, Е. В. Кремень. – Минск : БГУ, 2023. – 34 с.

Приводятся вопросы по алгоритмизации, представлению информации в памяти компьютера, базовому синтаксису C++, а также по операторам управления программой. Они предназначены для оценки и закрепления теоретических знаний по соответствующим темам.

УДК 004.42(075.8)
ББК 32.973-018я73-1

© Кравчук А. С., Кравчук А. И.,
Кремень Е. В., 2023
© БГУ, 2023

Оглавление

Введение.....	4
Алгоритмизация	5
Представление информации в памяти компьютера	12
Общие вопросы C++. Неименованные константы	14
Операции C++	17
Типы и модификаторы	25
Объявления	26
Операторы.....	28
Литература	34

Введение

Одним из направлений совершенствования процесса обучения является разработка системы контроля знаний, умений и навыков, позволяющей объективно оценивать знания обучающихся.

В настоящее время существует большое количество разнообразных способов проведения контроля и оценки знаний. Наиболее широко применяется тестирование как один из методов контроля усвоения обучающимися знаний по дисциплине, обладающий рядом определенных преимуществ перед традиционными методами контроля знаний (контрольная работа, устный ответ и т.д.).

Инструментом для измерения по шкале достижений обучающегося является, который соответствует не только предмету обучения, но и его задачам и служит развитию системного подхода к изучению учебной дисциплины.

Тестирование по методическому материалу из введения в любой предмет является крайне необходимым инструментом, побуждающим студентов активно включиться в образовательный процесс с первых занятий. Оно необходимо также с той точки зрения, что с течением времени из-за больших пробелов в базовых знаниях им будет все более сложно усваивать новый теоретический материал на необходимом уровне, что неизбежно приведет к низкому качеству полученных по предмету знаний.

Алгоритмизация

1. Основные этапы решения задачи?
 - постановка (формулировка) задачи;
 - построение модели, выбор метода решения задачи;
 - разработка алгоритма;
 - проверка правильности алгоритма;
 - анализ алгоритма и его сложности;
 - реализация алгоритма;
 - отладка программы, обнаружение, локализация и устранение возможных ошибок;
 - составление документации;
 - внедрение программы;
 - сопровождение программы;
2. Определение алгоритма?
 - это конечная совокупность точно заданных действий, для решения некоторой задачи;
 - алгоритм означает точное описание некоторого процесса (инструкцию по его выполнению);
 - нет правильного ответа;
3. Определение алгоритмизации?
 - это техника разработки алгоритма для решения поставленной задачи;
 - это техника составления плана для решения поставленной задачи;
 - нет правильного ответа;
 - процесс корректного преобразования алгоритма в программу на одном из языков программирования;
4. Определение понятия "реализация алгоритма"?
 - это техника разработки алгоритма для решения поставленной задачи;
 - это техника составления плана для решения поставленной задачи;
 - нет правильного ответа;
 - процесс корректного преобразования алгоритма в программу на одном из языков программирования;
5. Что такое отладка программы?
 - исправление синтаксических и логических ошибок;
 - процесс корректного преобразования алгоритма в программу на одном из языков программирования;
 - нет правильного ответа;

6. Определить понятие "определенность (детерминированность) алгоритма"?
- такое составление предписания, которое не оставляет места для различных толкований или искажений результата, т.е. последовательность действий алгоритма строго и точно определена;
 - возможность использования любых исходных данных из некоторого допустимого множества;
 - означает, что при любом допустимом исходном наборе данных алгоритм закончит свою работу за конечное число шагов;
7. Определить понятие "массовость алгоритма"
- такое составление предписания, которое не оставляет места для различных толкований или искажений результата, т.е. последовательность действий алгоритма строго и точно определена;
 - возможность использования любых исходных данных из некоторого допустимого множества;
 - означает, что при любом допустимом исходном наборе данных алгоритм закончит свою работу за конечное число шагов;
8. Определить понятие "результативность (конечность) алгоритма"
- такое составление предписания, которое не оставляет места для различных толкований или искажений результата, т.е. последовательность действий алгоритма строго и точно определена;
 - возможность использования любых исходных данных из некоторого допустимого множества;
 - означает, что при любом допустимом исходном наборе данных алгоритм закончит свою работу за конечное число шагов;
9. Выберите графический способ представления алгоритмов:
- словесно-формульное описание;
 - блок-схема;
 - алгоритмические языки;
 - операторные схемы;
 - псевдокод;
10. Чем характеризуется формульно-словесный способ записи алгоритма?
- описание осуществляется с помощью слов и формул;

- содержание последовательности этапов выполнения алгоритмов записывается на естественном профессиональном языке предметной области в произвольной форме;
- для описания алгоритмов используются схемы алгоритмов или блочные символы (блоки), которые соединяются между собой линиями связи;
- каждый этап вычислительного процесса представляется геометрическими фигурами (блоками);
- порядок выполнения этапов обычно указывается стрелками, соединяющими блоки;

11. Чем определяется графический способ описания алгоритма (блок-схема)?

- описание осуществляется с помощью слов и формул;
- содержание последовательности этапов выполнения алгоритмов записывается на естественном профессиональном языке предметной области в произвольной форме;
- для описания алгоритмов используются схемы алгоритмов или блочные символы (блоки), которые соединяются между собой линиями связи;
- каждый этап вычислительного процесса представляется геометрическими фигурами (блоками);
- порядок выполнения этапов обычно указывается стрелками, соединяющими блоки;

12. Определить понятие "алгоритмический язык".

- это формальный язык, специально разработанный для записи алгоритмов;
- формальный язык, предназначенный для записи компьютерных программ;

13. Что такое "алфавит алгоритмического языка"?

- набор используемых символов;
- система правил, по которым образуются конструкции языка;
- система правил, строго определяющая смысл и способ употребления каждой конструкции языка;

14. Что такое "синтаксис алгоритмического языка"?

- набор используемых символов;
- система правил, по которым образуются конструкции языка;
- система правил, строго определяющая смысл и способ употребления каждой конструкции языка;

15. Что такое "семантика алгоритмического языка"?

- набор используемых символов;
- система правил, по которым образуются конструкции языка;
- система правил, строго определяющая смысл и способ употребления каждой конструкции языка;

16. Что такое "псевдокод"?

- система команд абстрактной машины;
- способ записи алгоритма с помощью операторов близких к алгоритмическим языкам;
- система правил, строго определяющая смысл и способ употребления каждой конструкции языка;

17. Какие типы алгоритмических процессов вы знаете?

- линейные;
- разветвляющиеся;
- циклические;
- функциональные;
- структурные;

18. Определение линейного алгоритма?

- это такой алгоритм, в котором все действия выполняются последовательно друг за другом и только один раз;
- это такой алгоритм, в котором первичные и промежуточные данные не оказывают влияния на направление процесса вычисления;
- алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий;
- алгоритм, в котором содержится цикл;

19. Определение разветвляющегося алгоритма?

- это такой алгоритм, в котором все действия выполняются последовательно друг за другом и только один раз;
- это такой алгоритм, в котором первичные и промежуточные данные не оказывают влияния на направление процесса вычисления;
- алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий;
- алгоритм, в котором содержится цикл;

20. Определение цикла?

- последовательность команд, которая повторяется до тех пор, пока не будет выполнено заданное условие;
- обеспечивает однократное выполнение одной или другой последовательности команд в зависимости от истинности или ложности некоторого условия;

21. Что такое программирование?

- раздел информатики, область знаний об алгоритмах и программах, их свойствах, а также исполнителях алгоритмов и программ;
- процесс создания программного обеспечения, которое включает в себя: анализ; проектирование (при использовании процедурного программирования этот раздел является алгоритмизацией в узком смысле этого слова); кодирование и компиляцию; тестирование и отладку;
- кодирование алгоритмов на заданном языке программирования;
- разработка логической схемы для программируемых логических интегральных схем (ПЛИС);
- процесс записи информации в постоянном запоминающем устройстве;

22. Перечислите основные элементы структурного программирования?

- исполняемые в линейном порядке выражения;
- вызовы подпрограмм и/или функций;
- операторы условия;
- циклические операторы;
- класс;
- объект;
- инкапсуляция;
- наследование;
- полиморфизм;

23. Перечислите базовые понятия ООП?

- исполняемые в линейном порядке выражения;
- вызовы подпрограмм и/или функций;
- операторы условия;
- циклические операторы;
- класс;
- объект;
- инкапсуляция;

- наследование;
 - полиморфизм;
24. Каким знаком на блок-схемах обозначается занесение значения в ячейку памяти переменной (операция присвоения)?
- <-
 - ->
 - =
 - :=
25. Какое максимальное число логических условий может содержать разветвляющийся алгоритм?
- любое конечное число условий;
 - одно;
 - пять;
26. Какое минимальное число ветвей обработки данных имеет разветвляющийся алгоритм?
- две;
 - три;
 - одну;
27. К какому типу алгоритма относится блок-схема нахождения корней квадратного уравнения?
- линейный;
 - циклический;
 - разветвляющийся;
28. К какому типу алгоритма относится определение попадает ли точка внутрь круга, если радиус круга известен, а его центр и точка задаются своими координатами?
- линейный;
 - циклический;
 - разветвляющийся;
29. Пусть даны два числа. К какому типу алгоритмов относится вычисление среднего арифметического кубов этих чисел и среднего геометрического модулей этих чисел?
- линейный;
 - циклический;
 - разветвляющийся;

30. К какому типу алгоритма относится вычисление расстояния между двумя точками с заданными координатами?

- линейный;
- циклический;
- разветвляющийся;

31. К какому типу алгоритма относится нахождение площади треугольника со сторонами по формуле Герона?

- линейный;
- циклический;
- разветвляющийся;

32. Пусть даны x , y , z к какому типу алгоритма относится нахождение w по произвольной формуле без каких-либо условий.

- линейный;
- циклический;
- разветвляющийся;

33. Пусть даны два числа. К какому типу алгоритма относится вывод на экран первого числа, если оно больше второго, и вывод обоих чисел, если это не так?

- линейный;
- циклический;
- разветвляющийся;

34. К какому типу алгоритма относится нахождение наименьшее из трех данных чисел a , b , c ?

- линейный;
- циклический;
- разветвляющийся;

35. Даны три числа a , b , c , являющиеся длинами сторон треугольника. К какому типу алгоритма относится определение типа треугольника (равносторонний, равнобедренный, разносторонний)?

- линейный;
- циклический;
- разветвляющийся;

36. Пусть задано натуральное число n . К какому типу алгоритма относится нахождение суммы натуральных чисел от 1 до n .

- линейный;

- циклический;
- разветвляющийся;

37. Пусть задано произвольное x и натуральное n . К какому типу алгоритма относится возведение x в степень n без использования стандартной функции.

- линейный;
- циклический;
- разветвляющийся;

38. Пусть задано натуральное число n . К какому типу алгоритма относятся вычисление факториала $n! = 1*2*3*...*n$?

- линейный;
- циклический;
- разветвляющийся;

39. Определить понятие «императивное программирование».

- это синоним понятия «структурное программирование»;
- это синоним понятия «процедурное программирование»;
- это синоним понятия «объектно-ориентированное программирование».

Представление информации в памяти компьютера

1. Представить число $54.987_{(10)}$ в развернутом виде.

- $5*10_{(10)} + 4*10_{(10)}^0 + 9*10_{(10)}^{-1} + 8*10_{(10)}^{-2} + 7*10_{(10)}^{-3}$
- $5*10_{(10)}^2 + 4*10_{(10)}^1 + 9*10_{(10)}^0 + 8*10_{(10)}^{-1} + 7*10_{(10)}^{-2}$
- $5*10_{(10)}^0 + 4*10_{(10)}^{-1} + 9*10_{(10)}^{-2} + 8*10_{(10)}^{-3} + 7*10_{(10)}^{-4}$

2. Представить число $1011.011_{(2)}$ в развернутом виде.

- $1*10_{(2)}^3 + 0*10_{(2)}^2 + 1*10_{(2)}^1 + 1*10_{(2)}^0 + 0*10_{(2)}^{-1} + 1*10_{(2)}^{-2} + 1*10_{(2)}^{-3}$
- $1*10_{(2)}^2 + 0*10_{(2)}^1 + 1*10_{(2)}^0 + 1*10_{(2)}^{-1} + 0*10_{(2)}^{-2} + 1*10_{(2)}^{-3} + 1*10_{(2)}^{-4}$
- $1*10_{(2)}^4 + 0*10_{(2)}^3 + 1*10_{(2)}^2 + 1*10_{(2)}^1 + 0*10_{(2)}^0 + 1*10_{(2)}^{-1} + 1*10_{(2)}^{-2}$

3. Представить число $DE.7_{(16)}$ в развернутом виде.

- $D*10_{(16)}^1 + E*10_{(16)}^0 + 7*10_{(16)}^{-1}$
- $D*10_{(16)}^2 + E*10_{(16)}^1 + 7*10_{(16)}^0$
- $D*10_{(16)}^0 + E*10_{(16)}^{-1} + 7*10_{(16)}^{-2}$

4. Представить число $555.654_{(8)}$ в развернутом виде.
- $5 \cdot 10_{(8)}^2 + 5 \cdot 10_{(8)}^1 + 5 \cdot 10_{(8)}^0 + 6 \cdot 10_{(8)}^{-1} + 5 \cdot 10_{(8)}^{-2} + 4 \cdot 10^{-3}$
 - $5 \cdot 10_{(8)}^3 + 5 \cdot 10_{(8)}^2 + 5 \cdot 10_{(8)}^1 + 6 \cdot 10_{(8)}^0 + 5 \cdot 10_{(8)}^{-1} + 4 \cdot 10^{-2}$
 - $5 \cdot 10_{(8)}^1 + 5 \cdot 10_{(8)}^0 + 5 \cdot 10_{(8)}^{-1} + 6 \cdot 10_{(8)}^{-2} + 5 \cdot 10_{(8)}^{-3} + 4 \cdot 10^{-4}$
5. Указать результат в двоичной системе счисления выражения $10011.1 + 11.00111$
- 10110.10111
 - 10111.10111
 - 11110.10111
6. Указать результат в двоичной системе счисления $1111.0111 - 1.0001$
- 1110.011
 - 1101.0110
 - 1110.0111
7. Указать результат в шестнадцатеричной системе счисления $A23.F1 + 1.7$;
- A25.61;
 - A24.61;
 - A24.71;
8. Указать результат в шестнадцатеричной $1343.31 - D1.7F$:
- 1271.B2
 - 1261.B2
 - 1271.C2
9. Перевести число 111000111.11101 из двоичной системы счисления в шестнадцатеричную систему счисления.
- 1C7.E1
 - 1C7.D1
 - 1B7.E1
10. Перевести число 111011.101 из двоичной системы счисления в шестнадцатеричную систему счисления.
- 3B.A
 - 4B.A
 - 3B.B

11. Перевести число -111,101 из двоичной системы счисления в шестнадцатеричную систему счисления.
12. Перевести число -10000111011,101 из двоичной системы счисления в шестнадцатеричную систему счисления.
13. Перевести число A01C,1FF из шестнадцатеричной системы счисления в двоичную.
14. Перевести число 68,01FA из шестнадцатеричной системы счисления в двоичную.
15. Перевести число -B1,1001 из шестнадцатеричной системы счисления в двоичную.

Общие вопросы C++. Неименованные константы

1. Признак однострочного комментария?
2. Признак начала многострочного комментария?
3. С какого знака начинается препроцессорная директива?
4. Манипулятор перевода курсора при выводе на экран на новую строку?
5. Манипулятор, позволяющий выводить на экран логический (булев) тип?
6. Программа на языке C++ начинает свою работу:
 - с первой строки первой модуля программы;
 - с функции main();
 - с произвольного места, помеченного программистом директивой #begin;
7. В языке C++ целые десятичные литеральные константы представляют собой:
 - последовательность цифр от 0 до 7, начинающаяся с 0, например 011 или 0147;
 - последовательность шестнадцатеричных цифр 0-9 и A-F, перед которой стоит 0X или 0x, например 0xffff;
 - последовательность цифр от 0 до 9, не начинающуюся с 0, например 23, 2003;
8. В языке C++ целые восьмеричные литеральные константы представляют собой:
 - последовательность цифр от 0 до 7, начинающаяся с 0, например 011 или 0147;

- последовательность шестнадцатеричных цифр 0-9 и A-F, перед которой стоит 0X или 0x, например 0xffff;
- последовательность цифр от 0 до 9, не начинающуюся с 0, например 23, 2003;

9. В языке C++ целые шестнадцатеричные литеральные константы представляют собой:

- последовательность цифр от 0 до 7, начинающаяся с 0, например 011 или 0147;
- последовательность шестнадцатеричных цифр 0-9 и A-F, перед которой стоит 0X или 0x, например 0xffff;
- последовательность цифр от 0 до 9, не начинающуюся с 0, например 23, 2003;

10. В языке C++ вещественные десятичные литеральные константы представляют собой:

- последовательность цифр, разделенных точкой, не начинающуюся с 0, например 23.0 или 3.14;
- последовательность цифр, разделенных запятой, не начинающуюся с 0, например 23,0 или 3,14;
- последовательность цифр, содержащих мантиссу и показатель степени числа 10, например, 3e10, 5.12E-6;

11. Записать литеральную константу 72e02 с использованием точки для разделения целой и дробной частей.

12. Записать литеральную константу 72e-02 с использованием точки для разделения целой и дробной частей.

13. Обычно идентификаторы Index, INDEX и index обозначают:

- одну и ту же переменную;
- две различных переменных;
- три различных переменных;

14. В языке C++ основное отличие переменных от именованных констант состоит в том, что:

- именованной константе присвоить значение можно только при ее объявлении, переменной – в любом месте программы;
- переменная существует в памяти машины, именованная константа не существует;
- именованную константу можно инициализировать значением при создании, переменную нельзя;

15. Массивы в языке C++ могут быть:

- только одномерными;
- одномерными или двумерными;
- с любым количеством измерений;

16. Индексация массивов в языке C++ начинается:

- с единицы;
- с нуля;
- с любого индекса, определяемого программистом;

17. Если одномерный массив в языке C++ состоит из N элементов, то его индекс может принимать значения:

- от 0 до $N-1$;
- от 1 до N ;
- от 0 до N ;

18. С точки зрения языка C++ выражение является истинным, если:

- это выражение равно 0;
- это выражение не равно 0;
- это выражение равно 1;

19. С точки зрения языка C++ выражение является ложным, если:

- это выражение равно 0;
- это выражение не равно 0;
- это выражение равно 1;

20. В языке C++ имя массива без индекса эквивалентно:

- значению нулевого элемента массива;
- указателю на нулевой элемент массива;
- адресу нулевого элемента массива;
- последнему элементу массива;

21. Символ конца строки?

22. Знак и наименование директивы препроцессора для подключения заголовочных файлов?

23. Наименование заголовочного файла для подключения средств стандартного ввода/вывода?

24. Наименование заголовочного файла для подключения библиотеки функций работы со встроенным типом строк?

25. Наименование заголовочного файла для подключения библиотеки математических функций?

26. Текст инструкции `using` (оператора) для подключения пространства имен `std`?
27. В языке C++ символьные литеральные константы (символьные литералы) представляют собой:
- одиночный символ, заключенный в апострофы, например `'w'`, `'g'` или `'7'`;
 - последовательность символов, заключенную в двойные кавычки, например `"Это строка"`;
 - последовательность цифр, не начинающуюся с 0, например `23`, `2003`;
28. В языке C++ строковые литеральные константы (строковые литералы) представляют собой:
- одиночный символ, заключенный в апострофы, например `'w'`, `'g'` или `'7'`;
 - последовательность символов, заключенную в двойные кавычки, например `"Это строка"`;
 - последовательность цифр, не начинающуюся с 0, например `23`, `2003`;

Операции C++

1. Обозначение (знак) операции декремент?
2. Обозначение (знак) операции инкремент?
3. Знак составного присваивания (присваивание со сложением) ?
4. Знак составного присваивания (присваивание с умножением) ?
5. Знак составного присваивания (присваивание с вычитанием) ?
6. Знак составного присваивания (присваивание с делением) ?
7. Знак унарной операции отрицания?
8. Знак операции сравнения (равно ли) ?
9. Знак операции "не равно"?
10. Знак операции сравнения "больше"?
11. Знак операции "меньше или равно"?
12. Знак операции "меньше или равно"?
13. Результат на экране?

```
...  
int n = 0;  
cout<<!n;  
...
```

14.Результат на экране?

```
...  
int n = 15;  
cout<<!n;  
...
```

15.Результат на экране?

```
...  
int n = 9;  
cout<<boolalpha<<!n;  
...
```

16.Результат на экране?

```
...  
int n = 0;  
cout << boolalpha << !n;  
...
```

17.Результат на экране?

```
...  
double a=0, b = 1.5;  
cout << !(a + b);  
...
```

18.Результат на экране?

```
...  
double a=0, b = 1.5;  
cout << boolalpha << !(a + b);  
...
```

19. Аналитическое выражение для результата операции $s*!s$, для любого значения s любого типа?

20. Аналитическое выражение для результата операции $s*!!s$ для любого значения s любого типа?

21. С помощью тернарной операции вычислить модуль переменной.

22.Результат на экране?

```
...  
cout << 5 % 2 ? 0: 1;  
...
```

23.Результат на экране?

```
...
int a, b;
a = b = 3;
cout << !(a== b);
...
```

24.Результат на экране?

```
...
int a, b;
a = b = 3;
cout.setf(ios::boolalpha);
cout << !(a== b);
...
```

25.Результат на экране?

```
...
int x = 0;
cout << !sin(x);
...
```

26.Результат на экране?

```
...
cout << (5<<2);
...
```

27.Результат на экране?

```
...
cout << (5>>2);
...
```

28.Результат на экране?

```
...
int n = 2;
cout<< boolalpha<< n * (n % 2 ? 1: 0);
...
```

29.Результат на экране?

```
...
int n = 3;
cout<< n * (n % 2 ? 1: 0);
...
```

30.Результат на экране?

```
...  
int n = 2;  
cout << n * (!n) % 2 ? 1: 0);  
...
```

31.Результат на экране (без использования манипуляторов и флагов форматирования)?

```
...  
cout<< true;  
...
```

32.Результат на экране (без использования манипуляторов и флагов форматирования)?

```
...  
cout<< false;  
...
```

33.Результат на экране?

```
...  
double x = 2;  
cout << !(x*x + 1);  
...
```

34.Результат на экране?

```
...  
double x = 2;  
cout << !( x*x*x + 1);  
...
```

35.Результат на экране?

```
...  
cout << (6<<3);  
...
```

36.Результат на экране?

```
...  
cout << (7>>2);  
...
```

37.Результат на экране?

```
...
cout << (9%2)*(7 << 2 ? 1: 0);
...
```

38.Результат на экране?

```
...
cout << 3.14 * (28>>3 ? 1: 0);
...
```

39.Результат на экране?

```
...
cout << 7/4;
...
```

40.Результат на экране?

```
...
int i = 2;
cout << (--i ? --i : ++i);
...
```

41.Результат на экране?

```
...
int j = -1;
cout << (++j ? j++ : j--);
...
```

42.К каким типам операндов применима операция %?

- int
- long
- unsigned
- signed
- double
- float

43.К каким типам операндов применима операция >>?

- int
- long
- unsigned
- signed
- double
- float

44. Как называется операция, выполняющая действия над двумя переменными?
45. Как называется операция, выполняющая действие только над одной переменной?
46. Знак операция взятия адреса переменной?
47. Знак операции разыменования?
48. Результат на экране?

```
...  
int i = 9;  
cout<< ++i;  
...
```

49. Результат на экране?

```
...  
int i = 0;  
cout<< i--;  
...
```

50. Результат на экране?

```
...  
int n = 9;  
cout<< !(n++);  
...
```

51. Результат на экране?

```
...  
int n = 9;  
cout<< !(++n);  
...
```

52. Знак операции уточнения имени?

53. Знак операции косвенной адресации?

54. Результат на экране?

```
...  
int a;  
cout<< !(a = 13);  
...
```

55. Результат на экране?

```
...  
double x, y = 3;  
cout<<!(x = y - 3.);  
...
```

56.Результат на экране?

```
...  
int a = 2.1;  
cout<< !((float) a ? 1 : 0);  
...
```

57.Результат на экране?

```
...  
double x = 2.5;  
cout<<!((int) x ? 1 : 0);  
...
```

58.Результатом выполнения операции $5 * 6$ будет число

- 30;
- 30.0
- 0

59.Результатом выполнения операции $5.0 * 6$ будет число

- 30;
- 30.0
- 300

60.Результатом выполнения операции $9 / 2$ будет число

- 5
- 4
- 4.5

61.Результатом выполнения операции $7.0 / 2$ будет число

- 3
- 4
- 3.5

62.Результатом выполнения операции $17 \% 5$ будет число

- 2
- 12
- 7

63.Результат на экране?

```
...  
int x, y;  
x = 1;
```

```
y = 1;  
x = y++;  
cout<<x;
```

...

- 0
- 1
- 2

64.Результат на экране?

```
...  
int x, y;  
x = 1;  
y = 1;  
x = y++;  
cout<<y;
```

...

- 0
- 1
- 2

65.С помощью операции взятия адреса можно получить указатель

- на любую переменную;
- только на локальную переменную;
- только на глобальную переменную;

66.Имея указатель (адрес переменной):

- всегда можно получить хранящееся по этому указателю значение переменной;
- можно получить хранящееся по этому указателю значение переменной, только если она глобальная;
- можно получить хранящееся по этому указателю значение переменной, только если она локальная;

67.Для доступа к элементам структур или общедоступным элементам классов используется операция уточнения имени. Первым операндом операции должен быть:

- объект класса или структура;
- указатель на объект класса или указатель на структуру;
- переменная перечислимого типа;

68. Для доступа к элементам структур или общедоступным элементам классов используется операция косвенной адресации. Первым операндом операции должен быть:

- объект класса или структура;
- указатель на объект класса или указатель на структуру;
- переменная перечислимого типа;

Типы и модификаторы

1. тип целый?
2. тип символьный?
3. тип логический?
4. тип с плавающей точкой?
5. тип с плавающей точкой удвоенной точности?
6. модификатор типа знаковый?
7. модификатор типа беззнаковый?
8. модификатор типа длинный?
9. модификатор типа короткий?
10. Если в `main()` не будет указан тип возвращаемого значения, то какой будет подразумеваться?
11. В языке C++ базовый тип данных «`char`» предназначен для хранения:
 - целых чисел или символов;
 - вещественных чисел;
 - только символов;
12. В языке C++ базовый тип данных «`int`» предназначен для хранения:
 - символов;
 - положительных и отрицательных целых чисел;
 - вещественных чисел;
13. В языке C++ базовый тип данных «`double`» предназначен для хранения:
 - символов;
 - вещественных чисел;
 - целых чисел;

Объявления

1. Выбрать правильный формат объявления переменной:
 - модификатор модификатор тип ИмяПеременной;
 - модификатор тип ИмяПеременной;
 - тип ИмяПеременной;
 - модификатор ИмяПеременной;
 - модификатор модификатор тип ИмяПеременной;
 - модификатор тип ИмяПеременной;
 - тип ИмяПеременной;
 - модификатор ИмяПеременной.
2. Выбрать правильный формат объявления именованной константы:
 - const модификатор модификатор тип ИмяПеременной;
 - const модификатор тип ИмяПеременной;
 - const тип ИмяПеременной;
 - const модификатор ИмяПеременной;
 - const модификатор модификатор тип ИмяПеременной;
 - const модификатор тип ИмяПеременной;
 - const тип ИмяПеременной;
 - const модификатор ИмяПеременной;
3. Выбрать правильный формат прототипа функции:
 - модификатор модификатор тип ИмяФункции(список, формальных, параметров);
 - void ИмяФункции();
 - тип ИмяФункции(список, формальных, параметров);
 - модификатор ИмяФункции();
 - void ИмяФункции(список, формальных, параметров)
 - модификатор тип ИмяФункции();
 - ИмяФункции(список, формальных, параметров) ;
 - модификатор ИмяФункции();
4. Объявить указатель pointer на переменную целочисленного типа.
5. Объявить ссылку ref на переменную a целого типа.
6. Объявить «статический» целочисленный массив A размерностью n элементов.
7. Объявить «статическую» строку str длиной 30 символов
8. Объявить динамический массив A чисел с плавающей точкой размерностью n

9. Записать макроподстановку с именем NAME для целочисленного значения 5 с помощью директивы define

10. Объявлено перечисление:

```
enum Colors
```

```
{  
    YELLOW,  
    WHITE,  
    ORANGE
```

```
};
```

Какое значение имеет константа YELLOW?

11. Объявлено перечисление:

```
enum Animals
```

```
{  
    ANIMAL_PIG = -4,  
    ANIMAL_LION,  
    ANIMAL_CAT,  
    ANIMAL_HORSE = 6,  
    ANIMAL_ZEBRA = 6,  
    ANIMAL_COW
```

```
};
```

Какое значение имеет константа ANIMAL_CAT?

12. Указать правильные варианты операторов при **выделении** динамической памяти под одномерный массив а размерностью n.

13. Указать правильные варианты операторов по удалению ленты динамической памяти, выделенной под одномерный массив а из целых чисел.

14. Указать правильные варианты операторов по **удалению** динамической памяти, выделенной под двумерный массив а размерностью pxm.

15. Указать правильные варианты операторов по **выделению** памяти под двумерный массив а целых чисел размерностью pxm.

16. Указать правильное объявление указателя на функцию с прототипом double function(double).

17. Указать правильное объявление массива указателей на 5 функции с прототипом double function(double, double).

18. Каким значением по умолчанию, в перечислимых типах представляется первая из набора именованных констант?

19. При объявлении перечислимых типов:

- можно указать значения именованных констант явным образом;
- нельзя указать значения именованных констант явным образом;
- можно указать значения именованных констант явным образом, только если первая будет иметь значение 0.

20. При объявлении перечислимых типов различные именованные константы:

- обязательно будут иметь различные числовые значения;
- могут иметь одинаковые значения;
- могут иметь одинаковые значения, только если первая будет иметь значение 0.

21. Объявить строку `str` как «статический» массив символов и инициализировать его строковым литералом «Hello».

22. Объявить строку `str` как статический массив символов и инициализировать его посимвольно строковым литералом «Hello».

Операторы

1. Написать оператор, вычисляющий вещественное число y равное дробной части числа x с плавающей точкой.
2. Символ, который из выражения создает оператор?
3. Пустой оператор?
4. Результат на экране?

```
...
int i = 1;
switch(i)
{
    case 1: cout<< "один"; break;
    case 2: cout<< "два"; break;
    case 3: cout<< "три"; break;
}
...
```

5. Результат на экране?

```
...
int i = 1;
switch(i)
{
```

```
case 1: cout<< "один";
case 2: cout<< "два"; break;
case 3: cout<< "три"; break;
}
...
```

6. Результат на экране?

```
...
int i = 1;
switch(i)
{
case 1: cout<< "один"; break;
case 2: cout<< "два";
case 3: cout<< "три"; break;
}
...
```

7. Результат на экране?

```
...
int i = 9;
switch(i)
{
case 9: cout<< "сентябрь";
case 10: cout<< "октябрь";
case 11: cout<< "ноябрь";
default: cout<< false;
}
...
```

8. Результат на экране?

```
...
int i = 10;
switch(i)
{
case 9: cout<< "сентябрь";
case 10: cout<< "октябрь"; break;
case 11: cout<< "ноябрь";
default: cout<< false;
}
...
```

9. Результат на экране?

```
...
int a, b = 5;
if (b % 2) a = b / 2;
else a = 3 * b;
cout << a;
...
```

10.Результат на экране?

```
...
int a = 6, b = 2;
if (!(a % b)) cout << a * b;
cout << "Ничего";
...
```

11.Результат на экране?

```
...
int a = 5, b = 2;
if ((float(a) / b - a / b) > 0.7) cout << a / 2;
else cout << 2 * b;
...
```

12.Результат на экране?

```
...
int a = 5, b = 2;
if ((a / b) % 2) cout << a * a;
else cout << 3 * b;
...
```

13.Результат на экране?

```
...
int a = 5, b = 2, c = 3;
if (a > b && a > c) cout << a;
else if (b > a && b > c) cout << b;
else cout << c;
...
```

14.Результат на экране?

```
...
int a = 1, b = 5, c = -3;
if (a < b && a < c) cout << a;
else if (b < a && b < c) cout << b;
else cout << c;
...
```

15.Результат на экране?

```
...
int a = 1, b = 5, c = -3;
cout<< (a < b && a < c) ? a : (b < a && b < c) ? b: c;
...
```

16.Результат на экране?

```
...
int a = 1, b = 5, c = -3;
cout<< ((a > b && a > c) ? a : ((b > a && b > c) ? b: c));
...
```

17.Выбрать правильный формат оператора for().

18.Выбрать правильный формат оператора while().

19.Выбрать правильный формат оператора do while().

20.Указать конструкцию «бесконечный цикл».

21.Какое значение получит переменная y в результате выполнения участка программы

```
int x, y;
x = 0;
y = 0;
if (x) y = 1;
```

22.Какое значение получит переменная y в результате выполнения участка программы?

```
int x, y;
x = 0;
y = 0;
if (!x) y = 1;
```

23.Какое значение получит переменная y в результате выполнения участка программы?

```
int x, y;
x = 1;
y = 1;
while (x < 1) {
x = x + 1;
y = y + 1;
}
```

24. Какое значение получит переменная *y* в результате выполнения участка программы?

```
int x, y;  
x = 1; y = 1;  
while (x < 2) { x = x + 1; y = y + 1; }
```

25. Какое значение получит переменная *y* в результате выполнения участка программы?

```
int x, y; x = 1; y = 1;  
do { x = x + 1; y = y + 1; } while (x < 1);
```

26. Какое значение получит переменная *j* в результате выполнения участка программы?

```
int i, j; i = 1; j = 1;  
do {  
    i++; j+=1;  
} while (i <= 2);
```

27. Какое значение получит переменная *b* в результате выполнения участка программы?

```
int a, b;  
for(a = 0, b = 1; a < 3; a++) {  
    b = b * 2;  
}
```

28. Какое значение получит переменная *a* в результате выполнения участка программы?

```
int a, b;  
for(a = 0, b = 1; a < 3; a++) {  
    b = b * 2;  
}
```

29. Какое значение получит переменная *b* в результате выполнения участка программы?

```
int a, b;  
for(a = 0, b = 0; a < 3; a++) {  
    b = b + 2;  
}
```


30. Какое значение получит переменная `b` в результате выполнения участка программы?

```
int a, b;  
for(a = 0, b = 0; a < 3; b+=1, a++);
```

31. Какое значение получит переменная `b` в результате выполнения участка программы?

```
int a, b;  
for(a = 0, b = 1; a < 3; b *= 5, a++);
```

32. Выбрать правильную последовательность работы оператора `for(секция1; секция2; секция3) оператор;`

33. Оператор `break` служит для того, чтобы:

- прекратить выполнение содержащего его ближайшего цикла `while`, `do ... while`, `for` или условного оператора `switch`;
- прекратить выполнение текущей итерации содержащего его ближайшего цикла `while`, `do ... while` или `for`;
- немедленно завершить программу;

34. Оператор `continue` служит для того, чтобы:

- прекратить выполнение содержащего его ближайшего цикла `while`, `do ... while` или `for` или условного оператора `switch` и передать управление на следующий за прерванным оператор;
- немедленно передать управление на проверку условия продолжения выполнения ближайшего цикла `while`, `do ... while` или `for`;
- досрочно завершить программу;

Литература

1. Кравчук, А.С. Язык С++. Императивное программирование : учебные материалы для студентов специальности: 1-31 03 08 «Математика и информационные технологии (по направлениям)» / А.С. Кравчук, А.И. Кравчук, Е.В. Кремень ; БГУ, Механико-математический фак., Каф. веб-технологий и компьютерного моделирования. – Минск : БГУ, 2023. – 389 с.
2. Кравчук, А.И. Сборник лабораторных работ и примеров решения задач по алгоритмизации и программированию на языке Си / А.И. Кравчук, А.С. Кравчук - Минск: Технопринт, 2002. - 116 с.

Учебное издание

Кравчук Александр Степанович
Кравчук Анжелика Ивановна
Кремень Елена Васильевна

Тест

Введение в методы программирования

Учебные материалы
для студентов специальности 6-05-0533-07
«Математика и компьютерные науки
(по профилизациям)»

В авторской редакции

Ответственный за выпуск *Е. В. Кремень*

Подписано в печать 25.08.2023. Формат 60×84/16. Бумага офсетная.
Усл. печ. л. 2,09. Уч.- изд. л. 1,94. Тираж 50 экз. Заказ

Белорусский государственный университет.
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий № 1/270 от 03.04.2014.
Пр. Независимости 4, 220030, Минск.

Отпечатано с оригинал-макета заказчика
на копировально-множительной технике
механико-математического факультета
Белорусского государственного университета.
Пр. Независимости 4, 220030, Минск.