1. фффффыыыыыыыыыыыыыыыыыыыыыыы1. Алгоритм бинарного поиска - берем отсортированный массив, смотрим в середину, если не нашли там число, в зависимости от того, что в середине — ищем это число этим же методом либо в левой части, либо в правой, откидывая средний элемент..

2. Алгоритм линейного поиска- Поиск значения функции осуществляется простым сравнением очередного рассматриваемого значения (как правило, поиск происходит слева направо, то есть от меньших значений аргумента к большим) и, если значения совпадают (с той или иной точностью), то поиск считается завершённым..

3. Алгоритм определения количества разрядов в числе – каждую итерацию делем число на 10,и подсчитывем сколько раз мы делили на 10

4. Алгоритм поиска минимума и максимума в массиве - создаем массив и инициализируем его, после чего создаем переменную min и max которые равны первому элементу массива, если min < элемента массива то записываем в переменную min самый минимальный элемент массива, если же max > элемента массива то записываем в переменную max самый максимальный элемент массива

5. Алгоритм сортировки «пузырьком» - Алгоритм состоит из повторяющихся проходов по сортируемому массиву. За каждый проход элементы последовательно сравниваются попарно и, если порядок в паре неверный, выполняется обмен элементов. Проходы по массиву повторяются N-1 раз или до тех пор, пока на очередном проходе не окажется, что обмены больше не нужны, что означает — массив отсортирован. При каждом проходе алгоритма по внутреннему циклу, очередной наибольший элемент массива ставится на своё место в конце массива рядом с предыдущим «наибольшим элементом», а наименьший элемент перемещается на одну позицию к началу массива («всплывает» до нужной позиции как пузырёк в воде, отсюда и название алгоритма)..

1. 6. Алгоритм суммирования элементов массива - создаем переменную которая будет под считовать сумму элементов массива, создаем массив и инициализируем его, и зацикливаем(с значением количеством элементов массива) переменную которая использует сокращенную арифметическую форму (+=) к массиву который мы создали, выводим на экран переменную которая посчитала сумму элементов массива.

7. В чем отличия строкового массива от числового - в том что используется разный тип данных и на конце строкового массива имеется элемент (по умолчанию) ‘\0’

8. Виды алгоритмов – линейный, разветвляющийся, циклический, комбинированный

9. Все ли части цикла for обязательны - нет

10. Где всегда должны находиться параметры по умолчанию - в прототипе данной функции.

11. Где можно объявлять функции - в ходе выполнения программы, или даже в других проектах.

12. Группы типов данных - 1)Целочисленные2)Вещественные 3) Логические 4)Пользовательские

13. Для чего используется дескриптор ввода и вывода в консоли - чтобы обратиться к буферу ввода данных и экранным буферам своей консоли.

14. Для чего используется оператор break внутри цикла – для досрочного выхода из цикла.

15. Для чего используется оператор continue внутри цикла - команда continue переходит на следующую итерацию цикла.

16. Для чего используется оператор ветвления if - для проверки некоторых условий

17. Для чего используется функция kbhit() – для отслеживания нажата клавиша любая или нет

18. Для чего используются srand и time - функция srand() устанавливает исходное число для последовательности, генерируемой функцией rand() (В свою же очередь эта функция возвращает псевдослучайные числа.).

19. Для чего используются операторы объединения - для объединения переменных, элементов, данных.

20. Для чего необходимо использовать пространство имён std – для подключения всех стандартных библиотек С++(STL)

21. Для чего нужна библиотека <conio.h> - для подключения функции \_getch()

22. Для чего нужна библиотека <iostream> - для подключения cout,cin.

23. Для чего нужна библиотека <math.h> - для подключения pow(),sqrt()

24. Для чего нужна библиотека <time.h> - для подключения srand(),time()

25. Для чего нужна библиотека <windows.h> - для подключения графических объектов,

26. Если в функции есть переменная с таким же названием, что и глобальная, что произойдет - ошибки не будет.

27. Иерархия типов данных - double & int вот к каким типом данных приводит компилятор все отсальные типы данных касающийся этих категорий.

28. Как вычислить адрес любого элемента в массиве – с помощью указателя

29. Как двумерный массив располагается в памяти - последовательно

30. Как закрыть все процессы по указанному имени – системное администрирование

31. Как закрыть файл программно – fclose()

32. Как изменить кодировку строки – setlocale()

33. Как изменить цвет для всего окна консоли - SetConsoleTextAttribute() – литерал цвета от 16

34. Как изменить цвет для одной строки текста – SetConsoleTextAttribute()

35. Как можно зациклить выполнение программы – с помощью вечного цикла

36. Как можно остановить программу на несколько секунд - Sleeptime()

37. Как назначить новый заголовок для окна консоли - system("title (имя)");

38. Как одномерный массив располагается в памяти - последовательно

39. Как определить общий размер массива – sizeoff()

40. Как определить, нажата ли левая кнопка мыши –

41. Как определить, нажата ли правая кнопка мыши - RIGHTMOST\_BUTTON\_PRESSED

42. Как открыть файл программно – fopen()

43. Как очистить экран консоли – system(“cls”)

44. Как перевернуть строку – с помощью обратного цикла

45. Как перевести число в строку – NumberintoString()

46. Как передать массив в функцию – с пощью указателя на массив

47. Как перезагрузить компьютер через 2 минуты – системное администрирование

48. Как показать на экран текущее время и дату – asctime(timeinfo = localtime(&rawtime)))

49. Как получить все события, произошедшие в консоли - INPUT\_RECORD

50. Как получить дескриптор мышки - ReadConsoleInput

51. Как правильно называть переменные - переменная не должна начинаться с цифри, имя переменной может состоять из букв латинского алфавита и из цифр, и (\_), если я хочу использовать два слова в названии переменной то нужно ставить(\_),Переменные регистрозависимые.

52. Как привести строку к верхнему или нижнему регистру – Uppercase(),Lowcase()

53. Как произвести мгновенный выход их программы (назвать как минимум три программных способа) – exit(0),return(все формы);

54. Как производится преобразование типов в выражении – от меньшего к большему (к double & int)

55. Как просмотреть список ошибок – написано в TimeList

56. Как работает программа, если встречает оператор cin>> - ждет пока пользователь не введет что – нибудь, после введения записывает в переменную которая стоит после >>, если это не коректное значение то программа сломается(упадет), или будет работать не правильно.

57. Как работают циклы вложенные друг в друга - сначала выполняется цикл №2 после его завершения выполняется столько раз цикл №2 сколько написано в цикле №1 .

58. Как сгенерироваь число в заданном диапазоне – rand () % (max - min + 1) + min;

59. Как скрыть стандартный мигающий курсор - CONSOLE\_CURSOR\_INFO cci;

cci.bVisible = false;

cci.dwSize = 100;

SetConsoleCursorInfo(h, &cci);

60. Как сменить кодировку для всего проекта - SetConsoleOutputCP(1251);

61.Как создать или удалить папку программно – системное администрирование

62. Как узнать длину строки – strlen()

63. Как узнать текущую позицию курсора в файле -SEEK\_CUR

64. Как узнать, достигнут ли конец файла – по символу \’0’

65. Как установить курсор в заданную позицию – SetConsoleCursorPosition()

66. Как установить режим использования мышки - Event.MouseEvent.dwMousePosition.

67. Какая системная команда является аналогом оператора cout<< printf()

68. Какие Escape-последовательности вы знаете, опишите их действие - выполняют в потоке вывода форматирование, вывод или печать определенной информации. Например: \r – возврат каретки в начало строки,\n – новая строка,\t – горизонтальная табуляция,\v – вертикальная табуляция,\>> - двойные кавычки,\` - апостроф,\\ - обратный слеш,\0 – нулевой символ,\? – знак вопроса,\a – сигнал спикера компьюетра.

69. Какие виды операторов вы знаете – операторы ввода, вывода, условий, постусловий

70. Какие две функции используются для работы с файлом посимвольно – fopen(), fgetc()

71. Какие две функции используются для работы с файлом построчно – fclose(),fwrite(),fgetc()

72. Какие операторы используются для проверки условия – if()

73. Какие типы данных вам известны, опишите каждый из них - Они бывают:1)Целочисленные: char – 1 byte, short – 2 bytes, long – 4 bytes, long long – 8 bytes , int – равен машинному слову (4 bytes) но не больше типа long 2) Вещественные(дробные) float – 4 bytes, double – 8 bytes, long double – 10 bytes. 3) Логический – bool( true = 1 or false = 0) 4) Пользовательские (enum – перечисление, struct – структура , union - объединение ,class – класс используется в языке С++).

74. Какие файлы находятся в директории проекта – папки и файлы

75. Какие функции и операторы ждут реакции пользователя (назвать как минимум три) – cin >> , scanf()

76. Какой тип данных используется для объявления количества элементов массива – const int

77. Когда и кем был разработан язык Си – в стенах Bell Labs в период с 1969 по 1973 годы

78. Математические функции – это функции которые выполняют арифметические операции над аргументами

79. Можно ли в указатель с одним типом данных вписывать адрес переменной с другим типом данных - нет

80. Назначение ключевого слова break - остановка

81. Назначение ключевого слова const – неизменяемая(постоянная)

82. Назначение ключевого слова unsigned – специфекатор типа данных

83. Назначение функции strcat() – добавление к строки элементов

84. Назначение функции strcpy() – копирование одной строки в другую

85. ???Назовите два программных способа переименовать файл -

86. ???Назовите два программных способа удалить файл –

87. Напишите общий синтаксис каждого из известных вам циклов – while(условие(выражение)) тело цикла. for(выражение ; выражение; выражение) тело цикла

88. Напишите общий синтаксис объявления массива – тип данных имая массив [количество элементов массива]

89. Общий синтаксис объявления двумерного массива - тип данных имая массив [количество элементов массива] [количество элементов массива]

90. Общий синтаксис объявления функции – тип данных имя функции(тип данных аргументы) тело функции

91. Общий синтаксис оператора if – if(условие) тело условного оператора

92. Объясните понятия ложь / истина в языке Си – если ложь то 0, если правда то 1.

93. Объясните принцип работы оператора if – проверяет условие записанное () и в зависимости от этого выполняет тело условного оператора

94. Объясните принцип работы цикла do while – сначала происходит действия пока действует условие в цикле while

95. Объясните принцип работы цикла for – выполняються выражение которые записанны() потом тело цикла

96. Объясните принцип работы цикла while – выполняется условие цикла while а потом тело цикла

97. Объясните, почему нельзя напрямую осуществлять в консоли вывод кириллицы - потому что будут непонятные символы

98. Обязателен ли else - нет

99. Обязательно ли указывать return в функции и в каких случаях – нет, если функция имеет любой тип данных кроме void.

100. Оператор new – обращается к физической памяти компьютера

101. Операторы равенства – присваивают значения переменных.

102. Операторы сравнения - сравнивают значения.

103. Опишите способы инициализации массива при создании – в скобаках {} – это все нули, если же {} будет что то записано то это инициализируется, с помощью цикла можно инициализировать.

104. Опишите формы инициализации строкового массива – с помощью {} = , с помощью cin.getline()

105. Основные элементы блок-схем - условный, циклический, комбинированный

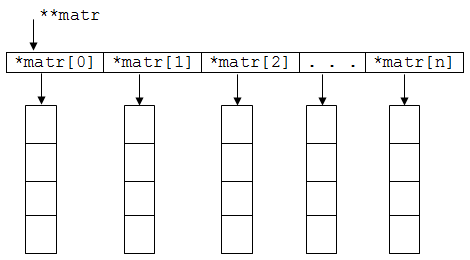
106. Особенности и недостатки в работе оператора cin>> - ждет пока пользователь введет значение, принимает все что введет пользователь

107. Особенности цикла for – можно не записывать выражения

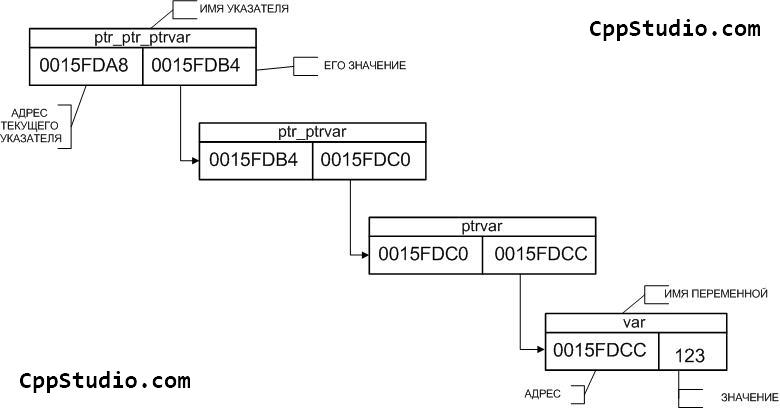
108. Отличие между компилятором и интерпретатором - компилятор проверяет правильность кода, а интерпретатор переводит код на языке программирование в машинный код понятный для процессора

109. Отличие функций strcmp() и stricmp() – 2 не регистрозависимая. Если совпадают – 0 , если 1 больше чем строка 2 - (-1), если 2 больше чем 1 - 1

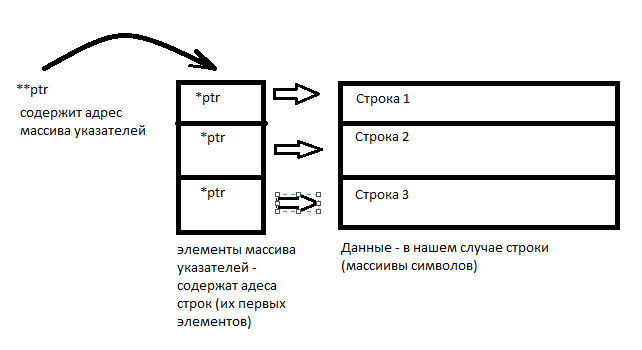
110. Отлов клавиатурных сообщений – производиться с помощью оператора cin>>

111. Отобразить графически создание двумерного динамического массива– (ГРАФИЧЕСКИ) 

112. Отобразить графически создание указателя -



113. Отобразить графически создание указателя на указатель



114. Перечислите коды основных клавиш – 27,72,80,75,74,15.

115. Перечислите операторы объединения – union , &&, &

116. По какой схеме работает логическое и – если 1 и 2 или больше условий правда переходим к дальнейшему выполнению кода программы

117. По какой схеме работает логическое или – если одно из условий правда переходим к дальнейшему выполнению кода программы

118. По какой схеме работает логическое отрицание – если условие не равно чему то переходим к дальнейшему выполнению кода программы

119. Понятие константы – это спецификатор который ставиться перед типом данных

120. Почему для стрелок нужно дважды ипользовать getch() – потому что имеют дополнительный код 224

121. Правила именования объектов – объекты могут иметь только ,символы подчеркивания, и цифры

122. Режимы открытия файла - r - только чтение. Запись запрещена. Файл должен существовать.

w - только запись. Чтение запрещено. Создает пустой файл. Если файл с таким именем существует,

то fopen его очистит до нулевой длины. (Всё содержимое файла будет потеряно, и он будет пуст)

a - запись в конец файла. Если файла не существует, он будет создан. Если файл существует, fopen откроет его

и ничего очищать не будет. Последующая запись будет дописана в конец файла.

r+ - открытие для чтения и записи. Файл должен существовать. Чтение начинается с начала файла.

w+ - открытие для чтения и записи. Создает пустой файл. Если файл с таким именем существует,

то fopen его очистит до нулевой длины. (Всё содержимое файла будет потеряно, и он будет пуст)

a+ - открытие для чтения и записи. Если файла не существует, он будет создан. Если файл существует, fopen откроет его

и ничего очищать не будет. Последующая запись будет дописана в конец файла.

123. С какой строчки компилятор начинает проверку исходного кода – с начало функции

124. С чего начинается нумерация элементов в массиве – c 0

125. Синтаксис объявления переменной – тип данных имя переменной

126. Сколько значений можно вернуть из функции за один раз - 1

127. Способы записи алгоритма - - вербальный, когда алгоритм описывается на человеческом языке;  
- символьный, когда алгоритм описывается с помощью набора символов;  
- графический, когда алгоритм описывается с помощью набора графических изображений.

128. Тонкости ввода, вывода строк – не забыть о \’0’

129. Указатель какого типа используется для хранения адреса файла в оперативной памяти – int;

130. Формы оператора return - Выполнение возобновляется в вызывающей функции в точке сразу после вызова.. Завершает выполнение функции и возвращает элемент управления в вызывающую функцию( или в операционную систему при передаче управления из функции main),return из функции,return из цикла.

131. Чем всегда заканчивается строковый массив – \’0’

132. Чем должна заканчивается большинство команд на языке С - ;

133. Чем инициализируется локальная переменная по умолчанию – значением(указателем)

134. Чем отличаются функции getch(), getc(), gets() и fgets() – 1) считывает нажата ли клавиша 2)возращает символ из потока filestream 3) считывает символы из стандартного потока ввода до символа новой строки 4) считывает символы из потока и сохраняет их в виде строки в параметр  в файле

135. Числа в каком диапазоне возвращает функция rand() в результате своей работы – целые

136. Что возвращает в результате своей работы функция time(0) -  возвращает текущее календарное время системы на 0

137. Что делает функция rewind() - Устанавливает внутренний указатель положения файла в начальное положение (начало файла).

138. Что значит операция взятия адреса и операция разыменования – 1)возвращает адресу в оперативной памяти2)возвращает переменной(указателя на переменную)

139. Что из себя представляет название функции без указания параметров – пустую функцию

140. Что нужно написать в коде, чтобы сгенерировалось число в диапазоне от 0 до 64 – rand() % 65

141. Что нужно написать в коде, чтобы сгенерировалось число в диапазоне от -15 до 0 – rand() % (max - min + 1) + min; где max = 0 а min = -15;

142. Что нужно написать в коде, чтобы сгенерировалось число в диапазоне от -48 до 11 - rand() % (max - min + 1) + min; где max = 11 а min = -48;

143. Что означает void – это означает что функция не будет иметь возвращаемого значения

144. Что подставляется на место вызова функции – указатель на функцию

145. Что произойдет, если return указывается в main - ошибка

146. Что произойдет, если при вводе значений в массив указать несуществующий индекс элемента и почему – не будет ошибки потому что элементы в массиве находятся в последовательно

147. Что произойдет, если при создании массива проинициализировать его большим количеством элементов, чем он задан – остальные недостающие элементы заполняться 0

148. Что происходит, если в функции срабатывает return – возвращает значение по оператору return

149. Что такое cin>> и cout<< - оператора ввода и вывода информации

150. Что такое EOF – конец файла

151.Что такое Escape-последовательность – это основные управляющие символы, служащие для командой для средства вывода.

152. Что такое return – это оператор который возвращает значение

153. Что такое void main() – это главная функция программы с которой компилятор начинает работу и которая не имеет возвращаемого значения так как поставлен тип данных этой функции void.

154. Что такое алгоритм – это список определенных последовательных действий которые в конце приводят к определенному результату

155. Что такое аргументы функции – это значение (число, указатель и т. д.), передаваемое функции, а также символьное имя (название переменной) в тексте программы, выступающее в качестве идентификатора этого значения..

156. Что такое библиотека – это подключаемый файл в наш проект

157. Что такое возвращаемое значение – это значение которое возвращает оператор return

158. Что такое генератор случайных чисел и по какому принципу он работает – это цикл который создает случайные числа в определенном интервале.

159. Что такое глобальная переменная, и чем инициализируется глобальная переменная по умолчанию – это переменная которая может быть использованная во всех блоках программы, константной переменной, которая живет на протяжении жизни всей программы.

160. Что такое двумерный массив – это одномерный массив, элементами которого являются одномерные массивы. Другими словами, это набор однотипных данных, имеющий общее имя, доступ к элементам которого осуществляется по двум индексам.

161. Что такое декремент и каковы его формы - это –- (уменьшение на 1) он бывает префиксный это когда операция указывается перед операндом и постфиксный это наоборот после операнда операция.

162. Что такое деление по модулю, опишите принцип и все тонкости его работы - это арифметическая операция которое записывается и вызывается в программе вот так %. Основной принцып – это сколько числе поместиться в указаном числе. Например : есть полная 50л канистра и 3л банка. И ты начинаешь вычерпывать из канистры банкой воду (набирать можно только полную банку). 48л вычерпал, осталось 2 литра. Это и есть остаток от деления 50 на 3 по модулю. Если же указанное число меньше чем остаток от деления то тогда ответом будет указанное число.

163. Что такое инкремент и каковы его формы - это ++ (увеличение на 1)он бывает префиксный это когда операция указывается перед операндом и постфиксный это наоборот после операнда операция.

164. Что такое интерпретатор - – это программа которая по этапно(по операторно, построчно) анализирует, обрабатывает и выполняет программный код по мере необходимости .

165. Что такое ключевое слово - это определенные команды, зарезервированы компилятором для обозначения типов переменных, класса хранения, элементов оператора(Вообщем для работы с переменными).

166. Что такое компилятор - Компьютерная программа, предназначенная для перевода программного кода с конкретного языка программирования в машинный язык целиком.

167. Что такое лесенка if else if – опишите принцип использования - опишите принцип использования – если первое условный оператор не выполнился переходим к следующему условному оператору(либо: Условия вычисляются сверху вниз. Когда обнаруживается истинное условие, то выполняется оператор, связанный с этим условием, а остальная часть конструкции игнорируется. Если не найдено ни одного истинного условия, выполняется оператор, соответствующий последнему else. Последний оператор else часто играет роль оператора, выполняемого по умолчанию, то есть, если все условия ложны, то выполняется оператор, соответствующий последнему else. Если последний оператор else отсутствует, то не выполняется никаких действий в случае ложности всех условий).

168. Что такое линковщик - инструментарий который производит компоновку(объектных модулей).

169. Что такое литералы, приведите примеры - это фиксированное значение которое невозможно изменить, у которого нет имени Примеры: Целочисленные = 0 – 9 со знаком + либо -, или без знака, также могут быть в  экспоненциальной  форме то есть е либо Е (-5,3Е), Строковые литералы – это \t, \\, \”,\n.

170. Что такое массив - это совокупность однотипных переменных имеющих общее имя и которые расположены в памяти последовательно.

171. Что такое многомерный массив - Определение многомерного массива должно содержать информацию о типе, размерности и количестве элементов каждой размерности..

172. Что такое неявное преобразование типов - происходит автоматически, если доступно для данных типов данных.

173. Что такое область видимости - это область программы в пределах которой идентификатор(имя) некоторой переменной продолжает быть связанным с этой переменной и возвращать ее значение, и в дальнейшим может быть изменен в ходе написания программы.

174. Что такое однострочный и многострочный комментарий - пояснения к исходному коду, находящиеся непосредственно внутри комментируемого кода. Многострочный комментарий имеет неограниченную длину и отмечается /\* \*/ . Однострочный комментарий отмечается в начале //.

175. Что такое операнд - какой-либо элемпент (число, буква и т. д)

176. ???Что такое ОЭВ -

177.??? Что такое ОЭК

178. Что такое параметры по умолчанию - в прототипе данной функции. Аргументы прототипа функции.

179. Что такое перегрузка функции - это несколько функций( 2 или больше) с одинаковыми именами, но различными параметрами.

180. Что такое переменная - это именованная область оперативной памяти предназначенная для хранения информации. Имеет следующие параметры: 1) Имя 2) Адрес- это порядковый номер байта, начиная с которого переменная хранится в памяти.3) размер 4) Тип данных 5) Значение 6) Доступность перезаписи.

181. Что такое преобразование типов – это переход одного типа данных к старшему типу данных

182. Что такое программирование – это создание программ на определенном языке программирования

183. Что такое пространство имён – это подключение стандартных библиотек к оперативной памяти

184. Что такое прототип функции – это сылка на функцию

185. Что такое расширяющее преобразование типов - происходит преобразование от меньшего к большему типу данных.

186. Что такое рекурсия и каков принцип её работы- определение части функции (метода) через саму себя, то есть это функция, которая вызывает саму себя, непосредственно (в своём теле) или косвенно (через другую функцию)

187. Что такое случайное число – это псевдослучайное число с генерируемое генератом случайных чисел

188. Что такое сокращенные арифметические формы - это комбинированные операторы. Например : \*=, /=, %=, -=,+= .

189. Что такое строка – это массив который состоит из символов

190. Что такое сужающее преобразование типов - может происходит с потерей данных при преобразовании значения старшего типа в младший тип данных

191. Что такое тернарный оператор, объясните принцип его действия - это логическое выражение x ? значение 1: значение 2 обязательно должен присутствовать одинаковый тип данных, работает также как и if else.

192. Что такое тип данных - это диапозон значений, которая может принимать переменная, а также правила связанные с обработкой этого значения. Они бывают:1)Целочисленные: char – 1 byte, short – 2 bytes, long – 4 bytes, long long – 8 bytes , int – равен машинному слову (4 bytes) но не больше типа long 2) Вещественные(дробные) float – 4 bytes, double – 8 bytes, long double – 10 bytes. 3) Логический – bool( true = 1 or false = 0) 4) Пользовательские (enum – перечисление, struct – структура , union - объединение ,class – класс используется в языке С++).

193. Что такое указатель на указатель - это адрес, который ссылается на другой адрес, а он, в свою очередь,  ссылается на адрес в памяти, где хранятся данные

194. Что такое указатель, напишите общий синтаксис объявления указателя -1) это переменная хранящая в себе адрес другой переменной 2) тип данных \* имя указателя (= значение);

195. Что такое унарные и бинарные операторы - унарный оператор – это оператор который принимает одно значение, а битовый оператор – это оператор который принимает два значения, они выполняют некоторые действия с этими принимаемыми значениями.

196. Что такое файл - именованный набор  байтов, который может быть сохранен на некотором накопителе.

197. Что такое функция - это специальный инструментарий кода, который позволяет повторяющейся фрагмент кода вынести за тело программы

198. Что такое цикл - это инструментарий предназначенный для организации многократного исполнения определенного кода(набора инструкций).

199. Что такое явное преобразование типов - происходит с явным применением операции.

200. Этапы разработки программы - 1. *Постановка задачи —* выполняется специалистом в предметной области на естественном языке (русском, английском и т. д.). Необходимо определить цель задачи, ее содержание и общий подход к решению. Возможно, что задача решается точно *(аналитически),* и без компьютера можно обойтись. Уже на этапе постановки надо учитывать эффективность алгоритма решения задачи на ЭВМ, ограничения, накладываемые *аппаратным и программным обеспечением* (АО и ПО).

2. *Анализ задачи и моделирование —* определяются исходные данные и результат, выявляются ограничения на их значения, выполняется формализованное описание задачи и построение (выбор) математической модели, пригодной для решения на компьютере.

3. *Разработка или выбор алгоритма решения задачи —* выполняется на осно­ве ее математического описания. Многие задачи можно решить различными способами. Программист должен выбрать оптимальное решение. Неточности в постановке, анализе задачи или разработке алгоритма могут привести к *скрытой ошибке —* программист получит неверный результат, считая его правильным.

4. *Проектирование общей структуры программы —* формируется модель решения с последующей детализацией и разбивкой на подпрограммы, определяется "архитектура" программы, способ хранения информации (набор переменных, массивов и т. п.).

5. *Кодирование —* запись алгоритма на языке программирования. Современные системы программирования позволяют ускорить процесс разработки программы, автоматически создавая часть ее текста, однако творческая работа по-прежнему лежит на программисте. Для успешной реализации целей проекта программисту необходимо использовать *методы структурного программирования.*

6. *Отладка и тестирование программы.* Под *отладкой* понимается устранение ошибок в программе. *Тестирование* позволяет вести их поиск и, в конечном счете, убедиться в том, что полностью отлаженная программа дает правильный результат. Для этого разрабатывается *система тестов —*специально подобранных контрольных примеров с такими наборами па­раметров, для которых решение задачи известно. Тестирование должно охватывать все возможные ветвления в программе, т. е. *проверять все ее инструкции,* и включать такие исходные данные, для которых решение *невозможно.* Проверка особых, *исключительных ситуаций,* необходима для анализа корректности. Например, программа должна отказать клиенту банка в просьбе выдать сумму, отсутствующую на его счете. В ответст­венных проектах большое внимание уделяется так называемой "защите от дурака" подразумевающей устойчивость программы к неумелому обращению пользователя. Использование специальных программ — *отладчиков,* которые позволяют выполнять программу по отдельным шагам, просматривая при этом значения переменных, значительно упрощает этот этап.

7. *Анализ результатов —* если программа выполняет моделирование какого-либо известного процесса, следует сопоставить результаты вычислений с результатами наблюдений. В случае существенного расхождения необ­ходимо изменить модель.

8. *Публикация* результатов работы, *передача заказчику* для эксплуатации.

9. *Сопровождение программы —* включает консультации представителей заказчика по работе с программой и обучение персонала. Недостатки и ошибки, замеченные в процессе эксплуатации, должны устраняться.