Среда программирования и директивы

1. **Что заключается между /* и */?**

- Комментарии
- код программы
- библиотечные функции

2. **Какой максимальный размер исполняемого .com-файла?**

- 64Кб.
- Зависит от аппаратной платформы
- 640Кб
- 16Мб

3. **Можно ли с помощью #define описывать макросы?**

- Да, любой длины
- Нет
- Только в C++
- Да, но только длиной в одну команду

4. **Для чего в программе C++ используется директива #include?**

- для подключения заголовочных файлов системы
- для формирования потоков вывода
- для объявления переменных

5. **Правильность вызова функций производится компилятором с помощью…**

- данных заголовочных файлов
- параметров первичной инициализации системы
- параметров первичной компиляции системы
- шаблонов потоковых данных

6. **В чем заключается суть компоновки программы?**

- в объединении нескольких объектных модулей программы в один исполняемый
- в переводе текстового файла в объектный модуль
- в подготовке программы к выполнению
- в проверке ошибок и выполнении программы

7. **Какую строку должен содержать исходный файл, который обращается к функции из стандартной библиотеки?**

- #INCLUDE <stdio.h>
- INCLUDE "stdio.h"
- #DEFINE lib <stdio.h>
- INCLUDE <stdio.h>

8. **Что такое #include?**

- Директива препроцессора
- Директива процессора
- Директива компилятора
- Оператор

9. **Для обнаружения синтаксических ошибок используется…**

- компилятор
- модулятор
- компаратор
- деструктор
- конструктор

10. **Для устранения синтаксических ошибок используется…**

- текстовый редактор
- компилятор
- модулятор
- деструктор
- конструктор

11. **Максимальный объем массива при работе в Borland C++ под управлением MS-DOS составляет…**

- 64 Кбайт
- 16 Кбайт
- 128 Кбайт
- 256 Кбайт

12. **Каково преимущество использования ключевого слова const вместо директивы #define?**

- константа, определенная с помощью const, имеет тип, и компилятор может проследить за ее использованием в соответствии с объявленным типом
  - к константе, определенной с помощью const, можно применить операции инкремента и декремента
  - константа, определенная с помощью const, доступна в других модулях программы

13. Данные, помещенные в стек первыми…

- будут извлечены из стека последними
  - не имеют индексного номера
  - имеют индекс, равный 0
  - будут первыми извлечены из стека

14. Чем стек FIFO отличается от LIFO?

- Типом очереди.
  - Размером
  - Скоростью работы
  - Типом хранимых данных

15. Модуль программы на С правильно называется:

- Модуль
  - Библиотека
  - Функция
  - Процедура

Функции ввода и вывода

16. Спецификация %5f определяет:

- Печать вещественного числа в поле шириной не более пяти позиций
  - Печать целого числа в поле шириной не более пяти позиций
  - Печать вещественного числа в поле шириной не менее пяти позиций
  - Печать 5 букв f

17. Какая функция позволяет реализовать механизм записи по одному символу в терминал?

- putchar
  - getchar
  - getc
  - printchar

18. Что производит строка программы: cout << "a=";?

- выводит символы на экран
  - заносит значения в память
  - формирует системный шаблон
  - удаляет символ с экрана

19. Каким образом организуется ожидание дейтвий пользователя, пока пользователь не наберет на клавиатуре какое-либо число и нажмет клавишу Enter?

- 50%*cin >> a;
- 50%*scanf("%d", &a);
  - -100%cout >> a;
  - -100%getchio >> a;
  - -100%cout<<a;

20. Что означает строка цифр до точки между % и символом преобразования в функции вывода printf?

- максимальная ширина поля
  - минимальная ширина поля
  - вывод цифр и %
  - вывод количества %, равного количеству цифр

21. С помощью какой командной строки можно организовать запись стандартного вывода в файл outfile, а не на терминал?

- prog<outfile
  - outfile \! prog
  - prog \! outfile
  - prog>outfile

○

22. Что означает символ преобразования c в функции вывода printf?

- аргумент рассматривается как отдельный символ
○ аргумент преобразуется в беззнаковую десятичную форму
○ аргумент преобразуется к десятичному виду
○ аргумент преобразуется в беззнаковую шестнадцатеричную форму (без лидирующих 0x)
○ аргумент является строкой

23. Что означает символ преобразования u в функции вывода print?

- аргумент преобразуется в беззнаковую десятичную форму
○ аргумент рассматривается как отдельный символ
○ аргумент преобразуется к десятичному виду
○ аргумент преобразуется в беззнаковую шестнадцатеричную форму (без лидирующих 0x)
○ аргумент является строкой

24. С помощью какой функции может быть организована остановка программы до нажатия какой-либо клавиши?

- getch();
○ wait();
○ stop();
○ break();

25. Функция ввода имеет вид:

- cin >>
○ io >>
○ bin >>
○ con>>

26. Описание функции getch() находится в заголовочном файле…

- conio.h
○ iostream.h
○ getio.h
○ stdio.h

27. Какая из этих функция предназначена для вывода в файл?

- fwrite
○ fprint
○ scanf
○ fscanf

28. Какая модель памяти и разрядность процессора использовались, если printf ("%p", &a) вывело 0xffff74df:cd88e5c0?

- Segmented, 32-bit.
○ Flat, 64-bit.
○ Flat, 32-bit.
○ Paged, 64-bit.
○ Paged, 32-bit.
○ Segmented, 64-bit.

29. Значение кода преобразования %lx приводит к выводу аргумента типа long в:

- в виде шестнадцатеричного числа
○ в виде восьмеричного числа
○ в виде десятеричного числа
○ в виде двоичного числа

30. Что означает знак минус между % и символом преобразования в функции вывода printf?

- *выравнивание преобразованного аргумента по левому краю его поля
○ выравнивание преобразованного аргумента по правому краю его поля
○ выравнивание преобразованного аргумента по центру его поля

31. Какая из этих управляющих последовательностей служит для перевода курсора в начало следующей строки

- \n
○ \e
○ \f
○ \r

○

32. sprintf(NAME, "TEMP%d", N); Что произойдет в результате выполнения этого кода?

- создание в NAME строки вида TEMPNNN, где NNN - значение N
- ○ создание в NAME строки вида NNN, где NNN - значение N
- ○ компилятор выдаст ошибку

33. При вводе данных с помощью функции scanf используется

- форматная строка
- ○ модульная строка
- ○ контекстная строка
- ○ объектная строка

34. Форматный ввод не позволяет вводить числовые значения в переменные типа

- *int
- ○ long
- ○ char
- ○ double

35. ?Системы программирования для DOS ориентированы на однобайтовую кодировку символьных данных на базе кодовых страниц

- ASCII
- ○ Unicode
- ○ MAC
- ○ CodBC

36. Значения символьных данных эквивалентны

- однобайтовым целым числам
- ○ вещественным числам
- ○ массивам числовых данных
- ○ массивам строковых данных

37. Какую спецификацию имеет функция printf для вещественных чисел?

- %f
- ○ %d
- ○ %s
- ○ %r

38. Первый элемент строки это:

- первый символ строки
- ○ имя строки
- ○ длина строки
- ○ имя массива, содержащего строку

39. Выберите выражение, которое выводит адрес скалярной переменной testvar:

- 50%cout <<&testvar;
- 50%printf ("%p", &testvar)
- ○ -100%cout <<testvar;
- ○ -100%cout <<"&testvar";
- ○ -100%cout <<"testvar";

40. Вы можете считывать ввод, который содержит несколько слов, используя:

- gets()
- ○ getc()
- ○ getch()
- ○ getchar()

41. Стандартная функция scanf:

- Считывает значение переменной;
- ○ Выводит переменную;
- ○ Инициализирует переменную;
- ○ Освобождает переменную;
- ○ Связывает значение переменной.

42. Спецификация преобразования определяется при помощи символа:

- %;
- ○ &;
- ○ ^;
- ○ $;
- ○ =.

43. Фрагмент кода: printf(“This\nis\na\nC\nprogram.\n”);

- Напечатает сообщение «This is a C program.» так, чтобы каждое слово располагалось на новой строке;
- Напечатает сообщение «This\nis\na\nC\nprogram.\n» в одну строку;
- Напечатает сообщение «This is a C program.» в две строки, причём первая закончится на C;
- Напечатает сообщение «This is a C program.» так, чтобы каждое слово располагалось в новой позиции табуляции;
- Напечатает сообщение «This is a C program.» в одну строку.

44. Фрагмент кода: printf("This\tis\ta\tC\tprogram.\n");

- Напечатает сообщение «This is a C program.» так, чтобы каждое слово располагалось в новой позиции табуляции;
- Напечатает сообщение «This\nis\na\nC\nprogram.\n» в одну строку;
- Напечатает сообщение «This is a C program.» в две строки, причём первая закончится на C;
- Напечатает сообщение «This is a C program.» так, чтобы каждое слово располагалось на новой строке;
- Напечатает сообщение «This is a C program.» в одну строку.

Типы, операции и выражения

45. typedef int (*PFI) (); Что произойдет в результате выполнения этого кода?

- создает тип PFI для"указателя функции, возвращающей значение типа int"
- создает тип int для"указателя функции, возвращающей значение типа PFI"
- компилятор выдаст ошибку

46. С помощью какого оператора в языке С можно вводить новые типы данных?

- typedef
- type
- type def

47. Что будет результатом выражения: sizeof(OBJECT)?

- целое, равное размеру указанного объекта в байтах
- целое, равное размеру указанного объекта в битах
- вещественное число, равное размеру указанного объекта в байтах
- адрес OBJECT

48. Какое описание переменной необходимо использовать чтобы она могла использоваться в файле2, если она определена в файле1?

- extern
- global
- main

49. Какой тип данных в С из нижеперечисленных имеет диапазон значений от 0 до 255?

- unsigned char
- char
- unsigned short
- short

50. Укажите объявления переменных целого типа:

- long l;
- int i;
- float f;
- double d;

51. Что обозначает применение long перед типом числовых данных:

- проводится удвоение размера типа
- Производится уменьшение размера типа в два раза
- Тип приводится к float

52. ?Какая строка введет целое число и число с дробной частью?

- scanf("%d %f",&a,&b)
- scanf("%cel %drob",&a,&b)
- scanf("Введите целое и дробное число %d %f",a,b)
- scanf("%c %d",&c,%d)

53. Что в языке С означает строка: void*t;?

- Переменная t - указатель на тип void
- Указатель t умножен на тип void
- Константа t - указател на тип void
- указатель t инициализирует тип void
- Эта строка не имеет смысла

54. В чем разница между char, unsigned char, signed char?

- char - знак определяется установкой по умолчанию, unsigned char и signed char - задают знак явно.
- Модификатор signed не влияет на работу

- ○ Модификатор unsigned не влияет на работу
- ○ Модификатор signed не существует для типа char.
- ○ Нет разницы

55. Что в языке С называется указателем?

- ● Переменная, содержащая адрес другой переменной
- ○ Определитель сегмента стека в оперативной памяти
- ○ Номер регистра данных
- ○ Номер ячейки оперативной памяти
- ○ Метка, определяющая передачу управления

56. В каком из перечисленных мест объявляются локальные переменные?

- ● В любом месте локальной функции
- ○ В файлах с расширением .h
- ○ Только перед телом функции main
- ○ В любом месте программы
- ○ В объявлении прототипа функции

57. В каком из вариантов объявляется переменная целого типа, а ее адрес сохраняется в другой?

- ● int a;int *b;b=&a
- ○ int &a;int *b;a=b
- ○ int *a;int b;b=*a
- ○ int &a;int &b;b=&a
- ○ int a,int b,b=a

58. В языке Си указателем называется:

- ● Переменная, содержащая адрес другой переменной
- ○ Система помощи пользователя
- ○ Определитель сегмента стека в оперативной памяти
- ○ Номер ячейки оперативной памяти
- ○ Метка, опередеяющая передачу управления

59. В каком порядке будут выполнятся операции (),+,?,&&,^,%

- ● (),%,+,^,?
- ○ (),?,^,%,&&,+
- ○ ?,(),+,&&,%,^
- ○ ^,&&,(),%,+,?
- ○ (),+,&&,?,%,^

60. Что производит с переменной операция i++?

- ● Прибавляет 1 после участия i в др. операции
- ○ Присваивает ей статус глобальной переменной
- ○ Прибавляет 1 до участия i и в др. операции
- ○ Заносит значения переменной i в первые 2 регистра
- ○ Выделяет переменной 1 дополнительный байт

61. x=1,y=0. После операций y=x++ -1

- ● x=2,y=0
- ○ x=2,y=1
- ○ x=2,y=2
- ○ x=1,y=2
- ○ x=0,y=1

62. Где содеражатся только логические операции?

- ● &&,>,>=,<,<=,==,||,!,!=
- ○ &,&&,?,??,<,>,<=,++,!=
- ○ >,<,!=,*=,=,|,/,!
- ○ ||,->,==,!=,?:,+=,?=

63. Чем определяется порядок следования операций в выражении?

- ● Круглыми скобками и приоритетом операций
- ○ Размером выражения и приоритетом операций
- ○ Приоритетом операций
- ○ Круглыми скобками
- ○ Размером выражения и объемом оперативной памяти

64. Что сделает a = b = c?

- ● Присвоит переменной b значение c, а затем присвоит a значение b.
- ○ Присвоит переменным a и b значение переменной c.
- ○ Присвоит переменной a значение переменной b, а переменной b значение переменной c.

- ○ Присвоит переменной a значение переменной b.
- ○ Присвоит переменной b значение переменной c.

65. Что сделает a ^= b ^= a ^= b, если a и b переменные типа int ?

- ● Поменяет местами значения переменных a и b.
- ○ Обнулит переменные a и b.
- ○ Присвоит переменным a и b максимально значение.
- ○ Инвертирует переменные a и b.
- ○ Инвертирует переменные a и b и поменяет их значения местами.

66. Каково назначение выражения x&1, если x - целое число?

- ● Проверяет переменную x на нечетность
- ○ Сдвигает все биты на 1 влево
- ○ Сдвигает все биты на 1 вправо
- ○ Присваивает x бинарную единицу
- ○ Помещает переменную x по адресу 1

67. Опишите действие программы:

```
#include <stdio.h>
#include <string.h>
#define S Hello, \0World"
int main(void){
return ( printf(S) == strlen(S)?1:0); }
```

- ● Выводит "Hello, " и возвращает 1. В случае невозможности вывода, вернет 0.
- ○ Выводит "Hello, \0World" и возвращает 0.
- ○ Выводит "Hello, World" и возвращает 0. В случае невозможности вывода, вернет 1.
- ○ Ничего не выводит. Сравнивает количество символов в строке с длиной строки записанной в буфер вывода.
- ○ Ничего не выводит. Завершается с кодом 0, если строка ненулевая, в противном случае, завершается с кодом 1.

68. Чему равно значение выражения ++i + ++i, если i = 5?

- ● 14
- ○ 13
- ○ 10
- ○ 11
- ○ 12

69. ?Что означает данный фрагмент int i = 1; int &t=i;

- ● 50%переменная t ссылается на переменную i
- ● 50%переменная i ссылается на переменную t
- ○ -100%переменная t передаёт значение переменной i и более не связана с ней
- ○ -100%i и t не связаны друг с другом

70. Что означает данный фрагмент float i = 1.0; int &t=i;

- ● i и t не связаны друг с другом
- ○ переменная t ссылается на переменную i
- ○ переменная i ссылается на переменную t
- ○ переменная t передаёт значение переменной i и более не связана с ней

71. Что означает режим записи "a+b"

- ● Добавить в конец двоичного файла или создать двоичный файл для чтения/записи
- ○ создать текстовый файл для записи
- ○ дописать текстовый файл
- ○ открыть только для записи

72. Какие из ниже перечисленных объявлений корректны?

- ● 33.333%int a[0];
- ● 33.333%int b[ ];
- ● 33.333%int c[];
- ○ -100%int d[*];

73. Укажите корректный вариант объявления массива

- ● 50%int array[3];
- ● 50%int array[];
- ○ -100%int array;
- ○ -100%int array[1..2]

74. Какие из перечисленных ключевых слов являются зарезервированными в C?

- 33.333%switch
- 33.333%enum
- 33.333%struct
- -100%undo

75. Имеется ли в языке Си булевский тип?

- Имеется в современной реализачии (_Bool)
- да, boolean
- нет

76. Истинным является выражение, значение которого...

- отлично от нуля
- равно TRUE
- равно 1

77. ?Что делают операции ++ и --?

- Увлививают/уменьшают значение операнда на 1
- Интерпретируются как + или -
- Увеличивают/уменьшают значение операнда
- Интерпретируются как + или -, но выполняются дважды

78. Чем отличается ++i от i++?

- 50%++i увеличивает значение переменной до ее использования
- 50%i++ увеличивает значение переменной после ее использования
- -100%++i увеличивает значение переменной после ее использования
- -100%i++ увеличивает значение переменной до ее использования

79. Выберите эквивалентную запись выражения "i = i + 5":

- i+=5
- i=+5
- i++5
- inc(i, 5)
- i add 5

80. К бинарным операциям относятся...

- 33.333%+, -
- 33.333%*, /
- 33.333%%
- -100%++, --
- -100%унарный -
- -100%+=, *=, /=...

81. Остаток от деления можно найти с помощью операции...

- %
- /
- mod
- div

82. Чему равно значение выражения "3/4"?

- 0
- 0.75
- 7.5E1
- 1

83. Чему равно число 015?

- 50%0xD
- 50%13
- -100%15
- -100%0.15
- -100%0xF
- -100%0x15

84. Что является символьной константой?

- Символ, заключенный в одинарные кавычки
- Символы, типа \t, \n, \\, \', \0...
- Символы вида \XXX, где X - восьмиричное число
- Символы, заключенные в двойные кавычки
- Все числа

- Символы, заключенные в кавычки /* ... */

85. Когда вычисляется константное выражение?

- На этапе компиляции
- На этапе линковки
- При выполнении

86. Как выглядят строковые константы?

- Последовательность символов, заключенных в двойные кавычки
- Последовательность символов, заключенная в одинарные кавычки
- Символы, находящиеся в символьном массиве
- Символы, заключенные в кавычки {* ... *}

87. Как помечается конец строки в памяти?

- По умолчанию, компилятор помещает в конец каждой строки \0
- По умолчанию, компилятор помещает в конец каждой строки LF
- По умолчанию, компилятор помещает в конец каждой строки CR-LF
- Длина строки храниться в первом (нулевом) байте строки

88. Чем отличаются записи "short int var" и "short var"?

- Ничем
- Базовым типом
- Второй вариант определяет вещественную переменную
- Второй вариант некорректен

89. Какие из объявлений имен не являются корректными?

- 33.333%1var
- 33.333%-var
- 33.333%return
- -100%_var

90. Какие, из перечисленных, типов относятся к базовым?

- 25%char
- 25%int
- 25%float
- 25%double
- -100%string
- -100%bool
- -100%byte

91. Сколько бит отводится под переменную типа long int?

- 50%не менее размера int
- 50%Зависит от архитектуры ЭВМ
- -100%32
- -100%48
- -100%64
- -100%от 32 до 64

92. Что делает модификатор unsigned?

- Явно указывает, что переменная беззнаковая.
- Явно указывает, что переменная содержит знак.
- Идентифицирует переменную общего назначения.
- Идентифицирует служебную (скрытую) переменную.

93. Какой тип будет иметь переменная, объявленная следующим образом: unsigned short int

- Беззнаковое целое короткое число.
- Короткое целое число со знаком.
- Короткое скрытое целое число.
- Беззнаковое число с плавающей точкой с пониженной точностью.
- Короткое целое число.

94. ?Допустимо ли использование комментариев в C++ программе?

- да, допустимо
- нет, не допустимо
- только в отдельном файле
- только в начале файлов

95. Какое значение будет присвоено переменной x при объявлении unsigned int x = -536?

- Нет правильных ответов
- -536
- 536
- 0

96. Чему равен результат выражения a---b? При a=8, b=6:

- 2
- 3
- 1
- 0

97. Оператор goto вызывает переход на:

- Метку
- Операцию
- Переменную
- Функцию

98. Операции && и ||:

- Комбинируют два булевых значения
- Сравнивают два численных значения
- Комбинируют два численных значения
- Сравнивают два булевых значения

99. Какого цикла не существует в языке C++

- Repeat
- For
- While
- Do

100. Как в языке программирования C++ обозначается логическое И

- &&
- ||
- |
- &
- and

101. Какая из этих операций имеет наивысший приоритет

- ++
- ==
- <=
- +=

102. Переменная, описанная внутри блока, видима:

- От точки своего объявления до конца блока
- От точки своего объявления до конца программы
- От точки своего объявления до конца функции
- Внутри функции

103. Значение 25%8 равно:

- 1
- 2
- 3
- 4

104. Как в языке программирования Си обозначается не равно:

- !=
- ><
- <>
- !==

105. Если язык обеспечивает возможность создания пользовательских типов данных, то говорят, что язык называется:

- Расширенным
- наследуемым
- инкапсулируемым
- перегруженным

106. Возможность выполнения оператором или функцией различных действий в зависимости от типа операндов называется:

- перегрузка
- полиморфизм
- инкапсуляция

- интеграция

107. Операция, выполняющая заданные действия над пользовательским типом данных, называется:

- перегруженной
- полиморфической
- инкапсулированной
- классифицированной

108. if (n > 0) { int i; for (i = 0; i < n; i++) ...}
Какова область действия переменной i?

- область ограниченная фигурными скобками
- область от первой фигурной скобки до конца программы
- весь файл
- функция main

109. Какую роль играют фигурные скобки в С?

- объединение описаний и операторов в составной оператор или блок
- разделитель операторов
- признак конца оператора
- разграничение комментариев

110. ?Каким образом записывается метка в С?

- имя метки, двоеточие
- двоеточие, имя метки
- имя метки, двоеточие, круглые скобки
- имя метки, двоеточие, фигурные скобки

111. A=5; B=8; if (A > B) Z = A; else Z = B;
Какое значение получит Z после выполнения этого кода?

- Z = 8
- Z = 5
- Z = 0
- Z = 12

112. Что означает оператор соотношения == в С?

- логическое равно
- побитовое равно
- не равно
- присвоить

113. Из каких символов не могут состоять имена переменных и символических констант?

- буквы a - я
- цифры 0 - 9
- буквы a - z
- знак подчеркивания

114. Какому выражению эквивалентна следующая запись: x *= y + 1?

- x = x * (y + 1)
- x = x * y + 1
- x += y * 1

115. Где в программе должны быть описаны переменные?

- до их использования
- в конце кода программы
- в С не описываются переменные
- их не обязательно описывать

116. Какой тип данных предусмотрен для чисел с плавающей точкой?

- float
- char
- int
- long

117. Что означает операция отношения != в С?

- не равно
- больше либо равно
- меньше либо равно
- логическое равно

118. ?Какой тип данных предусмотрен для символов?

- char
- int

- double
- float

119. Операция отношения:

- 50%имеет своим результатом булево значение
- 50%сравнивает значения двух операндов
- -100%присваивает значение одного операнда другому операнду
- -100%создает логическую комбинацию двух операндов

120. Напишите выражение, использующее операцию отношения, результатом которого является истина, если значения переменных george и sally не равны:

- george !=sally;
- george ==sally;
- george <>sally;
- george !! sally;

121. Напишите выражение с участием логической операции, принимающее истинное значение, если значение переменной limit равно 55, а значение переменной speed = 56:

- 25%limit == 55 && speed > 55;
- 25%limit = 55 && speed > 55;
- 25%limit == 55 & speed > 55;
- 25%limit = 55 & speed > 55;
- -100%Нет правильных ответов.

122. Перечислите в порядке убывания приоритетов типы операций:

- унарные, арифметические, отношения, логические, условные, присваивания
- унарные, отношения, логические, условные, арифметические, присваивания
- унарные, арифметические, условные, присваивания, отношения, логические
- унарные, арифметические, присваивания, отношения, логические, условные, присваивания

123. Выражение:

- 50%является частью оператора
- 50%всегда приводит к вычислению значения
- -100%является способом высказывания программы
- -100%всегда происходит вне функции

124. Какие из утверждений неверны:

- 50%операция взятия адреса & может применяться только к константам, выражениям и переменным, объявленным с модификатором register
- 50%указатель на void не может быть разыменован
- -100%единственное целое число, которое может быть присвоено указателю, это 0
- -100%указатели на разные типы данных не могут быть присвоены друг другу без использования преобразования типов.

125. Какие из утверждений верны:

- 50%единственное целое число, которое может быть присвоено указателю, это 0
- 50%указатели на разные типы данных не могут быть присвоены друг другу без использования преобразования типов.
- -100%операция взятия адреса & может применяться только к константам, выражениям и переменным, объявленным с модификатором register
- -100%указатель на void не может быть разыменован

126. Указатель - это:

- переменная для хранения адреса
- адрес переменной
- обозначение переменной, которая будет доступна следующей
- тип данных для адресных переменных

127. Если есть объявление int *test то выражение *test означает:

- значение переменной, на которую указывает test
- ссылку на значение переменной test
- разыменование переменной test
- ссылку на значение переменной, на которую указывает test
- указатель на переменную test

128. Тип переменной, на которую указывает указатель, должен присутствовать в определении указателя для того:

- чтобы компилятор мог правильно выполнять арифметические операции и получать доступ к элементам массива
- чтобы типы данных не перемешались при выполнении арифметических операций
- чтобы указатель мог быть использован для доступа к членам структуры
- чтобы не было затронуто ни одно из религиозных убеждений

129. Фрагмент кода вида: register int count = 0; объявляет целую переменную count, инициализированную значением 0, которая:

- Должна обрабатываться в регистре;
- Сохраняет своё значение между вызовами функции, в которой она определена;
- Является внешней, и чья область действия должна ограничиваться оставшейся частью файла, в котором она определена;
- Является внешней, и сохраняет своё значение между вызовами функции, в которой она определена;
- Должна обрабатываться в регистре и сохраняет своё значение между вызовами функции, в которой она определена.

130. Фрагмент кода:

```
int sum, x;
x=0;
Sum=0;
while (x<10) { Sum+=x; ++x;}
printf("%d", sum);
```

- Вычисляет сумму целых чисел от 0 до 9;
- Вычисляет сумму целых чисел от 0 до 10;
- Вычилсяет сумму целых чисел от 1 до 10;
- Вычисляет факториал числа x;
- Вычисляет факториал числа x-1.

131. Дано уравнение y=ax^3+7. Какой из перечисленных ниже операторов С соответствует данному уравнению:

- y=a*x*x*x+7;
- y=a*x*x*(x+7);
- y=(a*x)*x*(x+7);
- y=a*x*(x*x+7);
- y=a*x*x+7.

132. Несколько операторов, заключённых в фигурных скобках ({…}) называются:

- Составной оператор;
- Процедура;
- Функция;
- Последовательность операндов;
- Управляемая последовательность.

133. Операция x*=d+b - это:

- x=x*(d+b);
- x=x*d+b;
- x=x+d*b;
- x=d+b*x

Циклы и ветвления

134. В каком порядке необходимо располагать ветви case и default оператора switch

- Можно в любом
- Сначала все ветви case, потом default
- Сначала default, потом case
- Зависит от компилятора

135. Конструкция if(!valid) эквивалентна конструкции

- if (valid==0)
- if (valid!=NULL)
- if (valid==EOF)
- нет правильного ответа

136. Оператор break производит выход:

- Из текущего цикла или ветвления
- Только из цикла наибольшей глубины вложенности
- Только из ветвления switch наибольшей глубины вложенности
- Из всех вложенных циклов и ветвлений

137. Какого цикла не существует в языке Си++

- Repeat

- For
- While
- Do

138. Что происходит при использовании break в конструкции switch?

- выход из переключателя
- немедленное выполнение какого-либо оператора
- блокируется префикс default
- завершение программы

139. В каком случае точки с запятой могут отсутствовать в конструкции for?

- такого случая не может быть
- если какое-либо из трех выражений будет опущено
- если все три выражения будут опущены
- если опущены два выражения

140. Что происходит при выполнении оператора break в операторах for, while и do?

- немедленному выходу из самого внутреннего охватывающего его цикла
- немедленный выход из программы и выключение машины
- немедленному выходу из внешнего охватывающего его цикла
- выход из программы

141. Какой оператор дает способ выбора одного из вариантов, который заключается в проверке совпадения значения данного выражения с одной из заданных констант и соответствующем ветвлении?

- 50%switch
- 50%if
- -100%for
- -100%while

142. Какое положительное качество имеет отсутствие break в конструкции switch?

- возможность связать несколько случаев с одним действием
- расщепление при модификации программы
- уникальная переносимость на различных машинах
- экономное использование памяти

143. while (выражение) оператор; В каком случае эта конструкция будет выполняться бесконечно?

- если выражение всегда отлично от нуля
- если выражение всегда будет 0
- с помощью оператора while нельзя организовать бесконечный цикл
- зависит от среды разработки

144. for (выражение 1; выражение 2; выражение 3) оператор; Как записать тот же самый код с использованием конструкции while?

- выражение 1; while (выражение 2) { оператор; выражение 3; }
- выражение 3; while (выражение 2) { оператор; выражение 1; }
- выражение 2; while (выражение 1) { оператор; выражение 3; }
- этот код нельзя записать с использованием конструкции while

145. Какая конструкция языка С позволяет проверять условие окончания в конце, после каждого прохода через тело цикла?

- do while
- for
- while
- if

146. Что содержит вторая часть оператора for?

- условие, которое управляет циклом
- команда, которая выполняется один раз перед входом в сам цикл
- шаг реинициализации
- ничего

147. ?При использовании оператора while, в каком случае выполняется тело цикла?

- если истинно условие в круглых скобках
- если ложно условие в круглых скобках
- в любом случае выполняется
- в любом случае не вополняется

148. В цикле for, тело которого состоит более чем из одного оператора, точка с запятой ставится после:

- 50%каждого оператора в теле цикла
- 50%условия продолжения цикла
- -100%оператора цикла for
- -100%закрывающей фигурной скобки, ограничивающей тело цикла

149. Оператор break производит выход

- из цикла или ветвления switch
- только из цикла наибольшей глубины вложенности
- только из ветвления switch наибольшей глубины вложенности
- из всех вложенных циклов и ветвлений

150. Оператор goto вызывает переход на:

- метку
- операцию
- переменную
- функцию

151. В повторении, управляемом счётчиком, для подсчёта числа повторений группы команд используется:

- Контрольная переменная;
- Метка;
- Структура выбора;
- Оператор break;
- Оператор count.

152. В структуре повторения вызывает немедленное выполнение следующей итерации цикла оператор:

- Continue;
- Break;
- Goto;
- Exit;
- Skip.

153. Какие значения будут выведены: for (x=2; x<=13; x+=2) printf("%d", x);

- 2, 4, 6, 8, 10, 12;
- 4, 6, 8, 10, 12;
- 4, 6, 8, 10, 12, 14;
- 2, 4, 6, 8, 10, 12, 14;
- 2, 6, 10, 14.

154. Оператор принятия решений:

- If;
- Switch;
- Case;
- With;
- While.

Функции

155. Что возвращает malloc в случае неудачи?

- NULL
- Ничего
- -1
- 1

156. Что возвращает функция free()?

- Ничего
- Размер освобожденной памяти при успехе и NULL при неудаче
- NULL при неудаче и 1 при успехе
- Всегда возвращает 0

157. Перечислите 5 основных функций, динамически выделяющих память, языка С:

- allocmem, calloc, farmalloc, malloc, realloc
- allocmemory, free, malloc, farmalloc, hugealloc
- allocmemory, freemem, malloc, alloc, memoryalloc
- getmemory, initmemory, malloc, farmalloc, hugealloc
- realloc, malloc, farmalloc, nearalloc, hugealloc

158. Какая функция из стандартной библиотеки вызывает завершение выполнения программы?

- exit()

- end()
- return(desktop)
- close()

159. Что означает 0 в качестве аргумента функции exit?

- успешное завершение программы
- неудачное завершение программы
- ничего не означает

160. Определение любой функции имеет вид:

- тип-результата имя-функции(объявление аргументов) {объявления и инструкции}
- тип-функции имя-функции(объявление аргументов) {объявления и инструкции}
- тип-результата main(объявление аргументов) {объявления и инструкции}
- тип-результата имя-функции(объявления типов) {объявления и инструкции}

161. Может ли функция вызываться не main функцией?

- Может
- Только в стандарте ANSI
- Только в стандарте K&R
- Не может

162. ?В какой строке аргументы функции main написаны верно?

- (int argc,char*argv[])
- (int argc[],int argv)
- (int *argc,int argv)
- (int argc[],char argv)

163. Сколько точек выхода может иметь функция?

- Определяется пользователем
- Одну-по достижению }
- Две-по достижению } или return
- Три-по достижению },return или exit

164. ?Что произойдет, если в программе две функции будут различатся только типом возращаемого значения?

- Будет выдано сообщение, что вызов функции неопределен
- Компилятор выдаст сообщение о синтаксической ошибке
- Никаких ошибок не произойдет
- Из-за переполнения стека программа может неожиданно завершить выполнение

165. Как называется функция очистки экрана в библиотеке conio.h?

- clrscr();
- newscr();
- clean();
- dewash();
- clsscr();

166. Что сделает while (malloc (1)); ?

- Будет запрашивать память, пока та не кончится.
- Запросит 1 байт из heap.
- Будет бесконечно долго запрашивать память из heap
- Ничего

167. Что делает функция exit(1)?

- полностью выходит из программы
- выходит из испоняемой фукции
- выходит из исполняемого цикла
- ожидает нажатие клавиши

168. Что возвращает функция main()?

- возвращает значение при рекурсивном вызове
- возвращает значение операционной системе
- возвращает переменную которая может быть использована в дальнейшем.
- ничего никогда не возвращает

169. Рекурсия это -

- возможность функции вызывать саму себя
- возможность вызывать любую функцию из текущей функции
- это такое состояние функции когда она вызывает сама себя в бесконечном цикле
- возможность передовать в функцию функцию

170. Какая функция используется для возврата из функции?

- return
- recharge
- reopen
- repack

171. Сколько функций может содержать C++ программа?

- множество функций
- не более одной
- только функцию main
- ни одной

172. Если у функции отсутствуют аргументы, то в круглых скобках у нее указывается…

- void
- null
- empty
- 0

173. Сколько вызовов системных функций всегда будет в программе, независимо от ее реализации?

- всегда 2
- по крайней мере один
- не менее двух
- 0

174. Что выполняется в первую очередь при компоновке программы?

- директивы define,include
- функция main
- приведение типов
- проверка классов

175. Библиотечная функция exit() предназначена для выхода из:

- Программы, в которой она содержится
- Цикла, в котором она содержится
- Блока, в котором она содержится
- Функции, в которой она содержится

176. Библиотечная функция getche()

- Возвращает символ с отображением на экран в случае нажатия какой-либо из клавиш
- Возвращает символ, в случае нажатия клавиши Enter
- Получает информацию о количестве введенных символов
- Не отображает символ на экране

177. После имени функции ставится:

- ()
- []
- {}
- Ничего

178. ?Что происходит с автоматическими переменными, когда функция вызывает себя рекурсивно?

- при каждом обращении образуется новый набор всех автоматических переменных, совершенно не зависящий от предыдущего набора
- автоматические переменные не меняются
- автоматические переменные увеличиваются на постоянную величину
- создается копия предыдущего набора

179. ?Каким образом передаются аргументы функций в C?

- по значению
- по ссылке
- по значению и по ссылке
- они не передаются

180. Когда заканчивает существовать локальная переменная?

- как только закончится выполнение соответствующей функции
- как только закончится выполнение программы
- при выключении машины
- перед началом выполнения программы

181. Что содержит третья часть оператора for?

- продолжение цикла
- команда, которая выполняется один раз перед входом в сам цикл
- условие, которое управляет циклом

- такой части нет

182. Как выглядит описание функции, возвращающей значение типа double?

- double function()
- function (double)
- double int function (double)
- function of double

183. Какими переменными являются аргументы внутри функций?

- 50%локальными по местоположению
- 50%глобальными для функции
- -100%внешними
- -100%общими

184. Что является областью действия для автоматической переменной, описанной в начале функции?

- та функция, в которой описано имя этой переменной
- все функции, в которых описано имя этой переменной
- область программы от точки, в которой она объявлена в исходном файле, до конца этого файла
- весь файл

185. int (*comp)(); Что означает это описание?

- comp является указателем на функцию, которая возвращает значение типа int
- comp является указателем на функцию, которая ничего не возвращает
- comp является функцией, возвращающей указатель на целые
- comp является функцией, которая возвращает указатель на массив целых чисел

186. exit(EXIT_SUCCESS); эквивалентно:

- exit (0);
- exit (1);
- exit (SUCCESS);
- exit (EXIT_OK);

187. Библиотечная функция exit() предназначена для выхода из:

- программы, в которой она содержится
- цикла, в котором она содержится
- блока, в котором она содержится
- функции, в которой она содержится

188. Напишите выражение, которое вызывает метод cheep() для объекта класса bird, являющегося 27-м элементом массива manybirds:

- manybirds [26].cheep();
- cheep().manybirds [26];
- manybirds [27].cheep();
- cheep().manybirds [27];

189. Переменная, которая известна только внутри функции, в которой она определена называется:

- Локальная переменная;
- Контрольная переменная;
- Параметр;
- Неопределённая переменная;
- Управляющая переменная.

190. Чтобы показать, что функция не возвращает значений или не содержит никаких параметров в заголовке используется ключевое слово:

- Void;
- Main;
- Int;
- Decline;
- Define.

191. Для генерации случайных чисел используется функция:

- Rand;
- Srand;
- Randomize;
- Random;
- Ranval.

192. Переменная, объявленная вне любого блока или функции, называется:

- Глобальная;

- Локальная;
- Наружная;
- Общая;
- Основная.

193. Чтобы локальная переменная в функции сохраняла своё значение между вызовами функции, она должна быть объявлена со спецификатором класса памяти:

- Static;
- Dynamic;
- Unchangeable;
- Define;
- Register.

194. Функция, которая вызывает себя непосредственно или косвенно называется:

- Рекурсивная функция;
- Функция самовызова;
- Связанная функция;
- Параллельная функция;
- Структурная функция.

195. Каждая программа на С начинает выполнение с функции:

- Main;
- Void;
- Include;
- Int;
- Mail;

196. Тело каждой функции начинается и заканчивается с:

- {…};
- (…);
- Begin…end;
- Main…return;
- /*…*/;

Файлы

197. Что представляет собой внутреннее логическое имя файла?

- Указатель на одномерный массив
- Совокупность символов с символом завершения строки в конце
- Количество памяти, выделяемое под внешнее имя файла
- Указатель на начало таблицы размещения файлов
- Определяемая пользователем совокупность символов

198. С помощью какой функции можно открыть файл в С?

- fopen
- fileopen
- openfile

199. Какой символ обозначает конец файла?

- Конец файла никак не обозначается
- CR-LF
- CR
- LF
- 0
- 255

200. Какого типа должна быть переменная fd? fd = open ("filename", O_RDONLY, 0);

- integer
- *integer
- FILE
- *FILE

201. Какого типа должна быть переменная fd? fd = fopen ("filename", "w");

- FILE
- *integer
- integer
- *FILE

202. Функция read (FileDescriptor, ptr, 4) возвращает ноль, в случае...

- Конца файла
- Ошибки чтения
- Успешного завершения

- если файл открыт для записи

203. Признаком начала следующей строки в текстовом файле является символ(ы):

- 33.333%LF (0xA)
- 33.333%CR (0xD)
- 33.333%CR-LF (0xD, 0xA)
- -100%0x20
- -100%0xCR
- -100%0xFF

204. Работа с директивой #define

- это директива которая определяет идентификатор и последовательность символов, которая будет подставляться вместо идентификатора каждый раз, когда он встретится в исходном файле.
- после записи необходимо ставит точку с запятой
- это верная запись #define имя_макроса = последовательность_символов

205. Что делает функция fread()?

- считывает данные из файла в буфер
- считывает буфер в файл
- выводит информацию из файла на экран
- проверяет файл на ошибки

206. ?Что делает функция fclose()?

- записывает в файл все данные, которые еще оставались в дисковом буфере и закрывает файл на уровне операционной системы.
- не записывает в файл данные буфера,а сразу закрывает файл на уровне операционной системы.
- записывает в файл все данные, которые еще оставались в дисковом буфере и закрывает файл на низком уровне.
- не записывает в файл данные буфера,а сразу закрывает файл на низком уровне.

207. Что делает функция fopen()

- открывает файл
- читает строку из файла
- закрывает файл
- стирает файл

208. С помощью какой команды осуществляется включение файлов?

- #INCLUDE "FILENAME"
- #DEFINE "FILENAME"
- #DEFINE 'FILENAME'
- #INCLUDE 'FILENAME'

209. В чём разница между потоком и файлом?

- устройство ввода вывода на высоком уровне называется потоком, а само устройство - файлом
- устройство ввода вывода на высоком уровне называется файлом, а само устройство - потоком
- доступ потока не осуществляется с выводом или вводом на устройство
- доступ файла не осуществляется с выводом или вводом на устройство
- нет никакой разницы

210. что такое текстовые потоки?

- последовательностиь символов
- набор байтов
- массив символов

211. Какое утверждение верно?

- В языке Си файлом может быть всё что угодно (дисковый файл, терминал или принтер)
- Все потоки и файлы одинаковы
- невозможно потоком связаться с устройством
- ко всем файлам возможен прямой доступ

212. Какие из перечисленных ключевых слов являются зарезервированными в C?

- 33.333%switch
- 33.333%enum
- 33.333%struct
- -100%undo

213. Какие из перечисленных ключевых слов не являются зарезервированными в C?

- undo
- switch
- enum
- struct

214. Какие из ниже перечисленных объявлений корректны?

- 50%int a[0];
- 50%int b[255];
- -100%int c[];
- -100%int d[*];

215. Укажите корректный вариант объявления массива

- int array[3];
- int array;
- int array[];
- int array[1..2]

216. Существует массив int a[10], необходимо обратиться к элементу (5) какой способ верен?

- a[4]
- (a+4)
- (a+5)
- ((int)a+2)
- a[5]

Массивы

217. Как в языке С объявить массив целого типа из 10 элементов?

- int A[10]
- float*B (10)
- int*B[9]
- int A{10}
- float B{10}

218. Как в языке С представляются многомерные массивы?

- в виде последовательности символов в памяти
- в виде совокупности многомерных ячеек памяти
- в виде линейной комбинации независимых переменных
- в виде особого одномерного массива
- в виде совокупности регистров стека

219. В каком варианте правильно объявлен массив из 10 элементов целого типа?

- int a[10]
- int *a[9]
- int *a[11]
- float a[10]
- void a[10]

220. Чем характеризируется любой одномерный массив?

- Именем массива и числом элементов
- Именем массива и числом параметров
- Числом элементов и аргументами
- Параметрами и аргументами
- Константами и указателями

221. К чему может привести использование очень больших локальных массивов?

- Вследствие переполнения стека программа завершится до начала работы
- Программа не будет работать в системе DOS
- Снизится скорость работы
- Компилятор не сможет собрать программу

222. Почему нельзя передать двумерный массив функции, ожидающей указатель на указатель?

- Потому что массив превратится в указатель на массив
- Потому что размер указателя всегда больше размера массива
- Потому что в данном случае не определен размер указателя
- Потому что размер двумерного массива будет больше размера указателя на указатель

223. Что вычислит оператор sizeof при попытке определить с помощью него размер массива, который передан функции в качестве параметра?

- Размер указателя на массив
- Количество элементов массива
- Размер массива указателей
- Размер указателя на массив указателей
- Разме массива указателей на указатель

224. Какие из этих пар выражений не эквивалентны?

- Все эквивалентны
- arr[0][0] и **arr
- vec[i] и *(vec+i)
- arr[i][j] и *(*(arr+i)+j)

225. Максимальный объем массива при работе в Borland C++ под управлением MS-DOS составляет

- 64 Кбайта
- 16 Кбайт
- 128 Кбайт
- 256 Кбайт

226. Можно ли использовать в С многомерные массивы?

- можно
- только двумерные и трехмерные
- только двумерные массивы
- нельзя

227. Сколько элементов целочисленного массива будет зарезервировано в памяти при объявлении int c[12]?

- 12
- 11
- 13
- объявление вызовет синтаксическую ошибку

228. Можно ли задать двумерный массив в динамической памяти?

- Да, выделив память для каждой строки в отдельности, и связав адреса строк в отдельном массиве.
- Да, используя "(user_t **) malloc (matrixSize)".
- Да, используя стандартную функцию matrix.
- Нет.

229. Какой размер в байтах будет иметь после инициализации массив int m[5] = {1, 2, 3, 4, 5, 6}?

- объявление вызовет синтаксическую ошибку
- 10
- 12
- 6

230. Выберите верные утверждения:

- 33.333% массив является группой логически связанных ячеек памяти
- 33.333% индекс может быть целым числом или целочисленным выражением
- 33.333% общий размер одномерного массива вычисляется: всего байт = размер типа в байтах * количество элементов
- -100% элементы одного массива могут принадлежать разным типам
- -100% индекс седьмого элемента массива равен 7

231. Объявите целочисленный массив с 3 строками и 3 столбцами, предполагая, что была определена символическая константа SIZE, равная 3:

- int table [SIZE][SIZE];
- double table [SIZE][SIZE];
- float table [3][3];
- char table [SIZE][SIZE];

232. Доступ к элементам массива осуществляется с помощью:

- индекса элемента
- подхода FIFO
- операции точки
- имени элемента

233. Напишите выражение, которое определяет одномерный массив, именованный как doubleArray, типа double, содержащий 100 элементов:

- double doubleArray [100];
- double = doubleArray [100];
- double &doubleArray [100];
- double[100] doubleArray;

234. Элементы 10-элементного массива нумеруются начиная с и до:

- 0,9
- 1,10

- неизвестно, зависит от структуры памяти
- неизвестно, зависит от разрядности процессора

235. ?Напишите выражение, которое выводит j элемент массива doubleArray с помощью cout и операции <<

- cout <<doubleArray [j];
- cout <<"doubleArray [j]";
- cout <<"%s"doubleArray [j];
- cout <<"%s",doubleArray [j];

236. При доступе к многомерному массиву его индексы:

- заключены в квадратные скобки
- разделены запятыми
- заключены в квадратные скобки и разделены запятыми
- разделены занятыми и заключены в квадратные скобки

237. Напишите выражение для доступа к 4-му элементу 2-го подмассива двумерного массива twoD:

- twoD [2][4];
- twoD [4][2];
- twoD [1][3];
- twoD [3][1];

238. Имя массива представляет собой:

- 50%ссылку на адрес в памяти
- 50%константный указатель
- -100%оператором
- -100%указателем на оператор

239. При передаче имени массива в функцию она:

- 50%работает с тем же массивом, с которым работает и вызывающая функцию программа
- 50%ссылается на массив, используя другое имя, чем то, которое используется в вызывающей программе
- -100%работает с копией массива, переданной программой
- -100%ссылается на массив, используя то же имя, которое используется в вызывающей программе

Структуры и классы

240. Что представляет собой полиморфизм в языке C++?

- Использование одного участка памяти многими функциями одновременно
- Иерархическая структура построения классов и структур
- Возможность доступа ко многим участкам памяти одновременно
- Возможность объявления в программе многих классов
- Возможность передачи свойств одного класса другому

241. Что представляет собой наследование в языке C++?

- Возможность передачи свойств одного класса другому
- Невозможность изменить члены класса вне класса
- Возможность изменить члены класса вне класса
- Использование одного участка памяти многими фукнкциями одновременно
- Возможность доступа членов класса к глобальным переменным

242. Что произойдет, если полю не присвоить имя?

- ничего, это допустимо, неименованные поля (только двоеточие и ширина) используются для заполнения свободного места
- ошибка при выполнении программы
- ошибка компилятора

243. Что такое объединение?

- это переменная, которая в различные моменты времени может содержать объекты разных типов и размеров, причем компилятор берет на себя отслеживание размера и требований выравнивания
- это набор из одной или более переменных, возможно различных типов, сгруппированных под одним именем для удобства обработки

244. Каким образом может быть осуществлен доступ к членам объединения?

- имя объединения.член
- указатель объединения -> член
- имя объединения -> член

245. С помощью какой конструкции может быть указан член определенной структуры?

- имя структуры.член
- член.имя структуры

- имя структуры:член

246. При обращении к полю структуры левым операндом операции (.) является:

- структурная переменная
- поле структуры
- имя структуры
- ключевое слово struct

247. Перечисление объединяет:

- 50%Именованные целые числа
- 50%Константные значения
- -100%Данные различных типов
- -100%Логически связанные переменные

248. Определение класса это:

- определение всех его методов и данных
- объявление всех его методов и полей
- инициализация всех его полей и вызов конструктора
- проверка всех его методов и полей

249. Пусть нам дан указатель p, указывающий на объект типа upperdass. Напишите выражение, позволяющее вызвать метод exclu() этого класса для данного объекта:

- 50%p->exclu();
- 50%(*p).exclu();
- -100%p->&exclu();
- -100%p->exclu();

250. Операция new:

- получает память для новой переменной
- возвращает указатель на переменную
- создает переменную с именем new
- позволяет узнать, сколько памяти свободно на данный момент

251. Как правильно описать структуру?

- после слово struct идет имя структуры, затем фигурные скобки, поля, а в конце точка с запятой
- после слово struct идет имя структуры, затем фигурные скобки, поля
- после слово struct идет имя структуры, затем круглые скобки внутри поля, а в конце без точки с запятой
- после слово struct идет имя структуры, затем описываются поля без точки с запятой, а в конце точка с запятой
- идет имя структуры, затем круглые скобки внутри поля, а в конце слово struct и точка с запятой

252. Как правильно обратиться к структуре? struct part{int modelnumber;int partnumber:float cost;}parti;

- parti.modelnumber = 6244;
- (*parti).modelnumber = 6244;
- &parti.modelnumber = 6244;
- parti->modelnumber = 6244;

253. Структура объединяет :

- данные разных типов
- переменные разных типов
- данные одного типа
- целые именованные значения

254. При обращении к полю структуры левым операндом операции (.) является:

- структурная переменная
- поле структуры
- имя структуры
- ключевое слово struct

255. ?Сколько байтов памяти займут три структурные переменные типа time, если структура time содержит три поля типа int, если int равен 4 байта?

- 12
- 24
- 18
- 36

256. Напишите выражение для доступа к переменной salary структуры, которая является 17-м элементом массива emplist:

- emplist [16].salary

- emplist [17].salary
- emplist.salary [16]
- emplist.salary [17]

257. Связный список - это:

- структура, в которой имеется элемент, представляющий собой указатель на следующий элемент
- структура, в которой каждый элемент состоит из данных или указателя на данные
- массив указателей, указывающих на элементы списки
- структура, в которой указатель указывает на следующие данные

258. Перечисление объединяет:

- именованные константные значения
- именованные целые переменные
- данные различных типов
- логически связанные переменные

259. Если мы хотим отсортировать множество больших объектов или структур, то будет более эффективным:

- создать массив указателей на них и отсортировать его
- поместить их в массив и сортировать как его элементы
- поместить эти объекты в связный список и отсортировать его
- поместить ссылки на эти объекты в массив и отсортировать его

Примеры

260.
```
#include <stdio.h>
void mystery(int a, int b)
{(!(a>b))?printf("First"):printf("Second");}
int main() {int a = 5, b = 4;mystery(a, b);return 0;}
```
Что будет выведено на экран?

- Second
- First
- 4
- 5
- ничего

261.
```
#include <stdio.h>
struct Node{
int i;Node* next; // 1
Node* prev; // 2
}node; // 3
int main()
{node.i = 5; // 4
node.next = NULL; node.prev = NULL; return 0; }
```
В каких строках содержатся ошибки?

- Нет ошибок
- 1
- 2
- 3
- 4

262. Что произойдет, если попытаться скомпилировать и запустить следующую программу?
```
#include <stdio.h>
int main(int argc, char *argv[])
{int a = 2;
switch(a)
{default:printf("default");break;
 case 1:printf("1");
break;}
return 0;}
```

- На экран будет выведено "default"
- Произойдет ошибка компиляции
- Произойдет ошибка времени выполнения
- На экран будет выведено "1"
- Программа будет успешно скомпилирована и ничего не выведет на экран.

263. В каких строчках кода есть ошибки?
```
const char *test() {
```

```
const char *s = "Hello" //1
", "  // 2
"world!"  // 3
;  // 4
return s;  // 5}
```

- тут нет ошибок
- 1
- 2
- 3
- 4
- 5

264. Сколько раз выполнится цикл?

```
int i;
while(i < 10) { printf("%d-ый раз\n", i+1); i++; }
```

- количество итераций не определено
- 9 раз
- 10 раз

265. Укажите корректный прототип для функции: int sum(int a, int b) {return a + b;}

- 33.333%*int sum(int, int);
- 33.333%*int sum(int a, int b);
- 33.333%*sum(int, int);
- -100%sum(int, int): int;
- -100%sum() int a, int b;

266. Что произойдет в результате компиляции и выполнения такого кода:

```
#include<stdio.h>
const int a = 12;
int c = a;      //1
int * b = &a;   //2
int main() {printf ("%d %d %d", a, *b, c+*b);return 0;}
```

- Произойдет ошибка компиляции в строке 2
- Произойдет ошибка компиляции в строке 1
- Будет напечатано: 12 12 24
- Возникнет ошибка выполнения

267. Что произойдет в результате компиляции и выполнения такого кода:

```
#include <stdio.h>
//const
int a = 12;
int c = a;   //1
int * b = &a; //2
int main () {printf ("%d %d %d", a, *b, c+*b);return 0;}
```

- Будет напечатано: 12 12 24
- Произойдет ошибка компиляции в строке 1
- Произойдет ошибка компиляции в строке 2
- Возникнет ошибка выполнения

268. В каких строчках содержатся ошибки?

```
char *s=”Hello”;  /* 1 */
char *p1=s, *p2;  /* 2 */
p2=s+4;  /* 3 */
printf(“%p”,p1 + p2);  /* 4 */
printf(“%p”,p2 – p1);  /* 5 */
printf(“%d”,(*p1 *p2);  /* 6 */
printf(“%p”,p2+*p1);  /* 7 */
printf(“%p”,p2-*p1);  /* 8 */
printf(“%d”,p2 – p1);  /* 9 */
```

- 4
- 1
- 2
- 3
- 5
- 6
- 7

- 8
- 9
- в этом коде нет ошибок

269. В каких строчках не содержатся ошибки?

```
char *s=”Hello”;  /* 1 */
char *p1=s, *p2;  /* 2 */
p2=s+4;  /* 3 */
printf(“%p”,p1 + p2);  /* 4 */
printf(“%p”,p2 – p1);  /* 5 */
printf(“%d”,(*p1 *p2);  /* 6 */
printf(“%p”,p2+*p1);  /* 7 */
printf(“%p”,p2-*p1);  /* 8 */
printf(“%d”,p2 – p1);  /* 9 */
```

- 2
- 3
- 5
- 6
- 7
- 8
- 9
- 4
- в этом коде нет ошибок

270. Что произойдет в результате компиляции и выполнения данного кода: #include <stdio.h>

```
void f() { printf("%d", 1); }
int main() {
void (*a)() = f; // 1
a();           // 2
return 0;}
```

- Будет напечатано число 1
- Ошибка времени выполнения
- Будет напечатано число 0
- Ошибка компиляции в строке 1
- Ошибка компиляции в строке 2

271. Что выведет на экран следующий участок кода, если по умолчанию тип char беззнаковый? char S[] = "EYY"; printf(S, S[printf("S")]--);

- SEXY
- EYYS
- YES
- SEYY
- EYES

272. Как в прототипе функции func() можно объявить параметр - одномерный целочисленный массив с именем x, чтобы следующий код скомпилировался?

```
int main(void){int i[10];func(i);return 0;}
```

- 33.333%*int *x
- 33.333%*int x[10]
- 33.333%*int x[]
- -100%Верного ответа нет

273. В каких строчках кода находятся ошибки?

```
#include <stdio.h>
int main() {if (0)  // 1
if (1){  // 2
for(;;);  // 3
while (1);}  // 4
else printf("foo");  // 5
else printf("bar");  // 6
return 0;}
```

- тут нет ошибок
- 1

- 2
- 3
- 4
- 5
- 6

274. Выберите все фрагменты кода, которые откомпилируются успешно (содержимое переменных и контекст применения значения не имеют)?

- 50%c = a+++ ++b;
- 50%c = a++ + ++b;
- -100%c = a++++ +b;
- -100%c = a+ ++++b;
- -100%c = a++ +++b;

275. Что выведет следующая программа?

```
#include <stdio.h>
int i1;
int main()
{static char c2 = 4;
i1 = 128;
printf("%d\n", c2 * (char)i1);
return 0;}
```

- Ошибка при линковке
- 0
- 512
- -512
- Зависит от компилятора

276. Что выведет следующая программа?

```
#include <stdio.h>
//extern
int i1;
int main()
{i1 = 128;printf("%d\n", (unsigned char) i1);getchar();return 0;}
```

- 128
- -128
- 512
- 0

277. Дан прототип функции: float calculate(int a, int b, float c); Как правильно объявить указатель на эту функцию и присвоить ему адрес функции?

- 50%float (*ptr)(int a, int b, float c) = calculate;
- 50%float (*ptr)(int, int, float) = calculate;
- -100%float* (ptr)(int, int, float) = calculate;
- -100%float* (ptr)(int, int, float) = calculate();
- -100%float (*ptr)(int, int, float) = calculate();

278. Какой будет результат следующей программы?

```
#include <stdio.h>
#include <stdlib.h>
int foo() {printf("foo ");return 1;}
int main() {int x = foo();printf("x = %d", x);return 0;}
```

- x = 1
- foo x = 1
- foo x = 0
- x = 0
- Ошибка во время выполнения
- Ошибка компиляции

279. Чему равен sizeof("ab\0cd\tefg")

- 10
- 13
- 9
- 3
- 20

280. Сколько параметров может принимать на вход данная функция: int foo_a();

- *любое
- 0
- 1
- 2

281. Что ошибочно в следующем фрагменте кода:
```
int * x = (int *) malloc(100 * sizeof(int));
x = realloc(x, sizeof(int) * 200);
```

- Если realloc не удастся, память будет утеряна
- Код корректен, так как realloc гарантированно произойдет без ошибок
- Код корректен, так как realloc изначально очистит память и даже в случае возникновения ошибки утечки памяти не будет
- Данный код не скомпилируется

282. Чему будет равно значение переменной x после выполнения следующего фрагмента кода:
```
int a = 010;
int b = 100;
int c = 110;
int d = 0x10;
int x = d / (c - b - a);
```

- 8
- 10
- 0
- 16
- 2
- Значение переменной x не определено

283. Что из перечисленного не могут возвращать функции?

- массивы
- функции
- фундаментальные типы
- указатели
- все возвращаемо

284. Допустима ли следующая конструкция: extern static int;

- Нет
- Да
- только при указании переменной
- да только в описании функции

Директивы препроцессора

285. Чем отличается записи #include <lib.h> и #include "lib.h"?

- 50%*#include <lib.h> подключает библиотеку из определенной реализацией директории.
- 50%*#inlude "lib.h" подключает библиотеку из той же директории, в которой находится исходный код программы.
- -100%#include <lib.h> подклбчает библиотеку из системной директории windows/include.
- -100%#inlude "lib.h" подключает библиотеку из директории, указанной компилятором.
- -100%никак

286. Что выполняет конструкция
```
#ifdef condition
...
#endif
```

- Если имя condition определено, то данный блок будет откомилирован. В противном случае блок будет пропущен.
- Если имя condition не определено, то данный блок будет откомилирован. В противном случае блок будет пропущен.
- Если условие condition истинно, то данный блок будет откомилирован. В противном случае блок будет пропущен.
- Если условие condition ложно, то данный блок будет откомилирован. В противном случае блок будет пропущен.

287. В чем ошибка?
```
#ifdef condition { ; } #endif
```

- Директивы препроцессора расположены в одной строке.
- Код в блоке условной компиляции записан в операторных скобках.
- Символ ; должен находиться за операторными скобками.

- Идентификатор condition не определен.
- Условие #ifdef должно быть записано в скобках.
- Нет ошибок.

288. В каких строках содержаться ошибки?
#define 1
#define A 2
#define B(3)
#define C(4) 4

- 33.333%*1
- 33.333%*3
- 33.333%*4
- -100%2
- -100%Во всех строках

289. #DEFINE external 02
#DEFINE static 04
Что означает следующая идиома: IF ((FLAGS & (external | static)) == 0) ... ?

- для проверки на включение битов external и static в FLAGS
- включает биты external и static в FLAGS
- выключает биты external и static в FLAGS

290. struct {
unsigned IS_KEYWORD : 1;
unsigned IS_EXTERN : 1;
unsigned IS_STATIC : 1; } FLAGS;
Как включить биты IS_EXTERN и IS_STATIC?

- FLAGS.IS_EXTERN = FLAGS.IS_STATIC = 1;
- FLAGS.IS_EXTERN = FLAGS.IS_STATIC = 0;
- FLAGS |= external | static;

291. В чем отличие конструкции #ifdef от #ifndef?

- 50%#ifdef-блок компилируется, если установлено условие.
- 50%#ifndef-блок компилируется, если условие не установлено.
- -100%#ifndef, от ifNewDefine - директива только C++ компилятора
- -100%#ifndef совмещает в себе конструкцию #define и #ifdef
- -100%Нет различий

292. Что делает следующий код?
**#define A **
int var = 3;

- Объявляет A как "int var = 3;"
- Объявляет A как "\" и объявляет переменную целого типа.
- Объявляет A как "\ int var = 3;"
- Компиляция данного кода вызовет ошибку.

293. Можно ли отменить действие #define?

- Да, с помощью конструкции #undef
- Да, с помощью конструкции #undefine
- Да, с помощью повторного использования конструкции #define
- Нет.

294. Какая ошибка допущена при объявлении макроса?
#define max(A,B) (A > B ? A : B)

- Нет ошибки
- Имена A и B должны стоять в скобках
- Выражение должно быть указано в фигурных скобках, а не круглых.
- Имена A и B не объявлены