

1. Что напечатает следующая программа

```
#include <stdio.h>
int i=0, j=2;
int main(void)
{
printf("i=%d j=%d \n",i,j);
return 0;
}
```

1. i=0 j=2
2. i=0.0 j=2.00
3. i=2 j=2
4. i=0 j=0
5. i=0 j=3

2. Что напечатает следующая программа

```
#include <stdio.h>
int i=0, j=2;
int main(void)
{
i+=1; j+=2;
printf("i=%d j=%d \n",i,j);
return 0;
}
```

1. i=0 j=2
2. i=1 j=4
3. i=2 j=2
4. i=0 j=0
5. i=0 j=3

3. Что напечатает следующая программа

```
#include <stdio.h>
int i=0, j=2;
int main(void)
{
i++; j++;
printf("i=%d j=%d \n",i,j);
return 0;
}
```

1. i=0 j=2
2. i=1 j=4
3. i=1 j=3
4. i=0 j=0
5. i=0 j=3

4. Что напечатает следующая программа

```
#include <stdio.h>
float i=2, j=4;
int main(void)
{
printf("i=%e j=%e \n",i,j);
return 0;
}
```

1. i=0 j=2
2. i=1 j=4
3. i=2.000000 j=4.000000
4. i=2.000000e+00 j=4.000000e+00
5. i=0 j=3

5. Что напечатает следующая программа

```
#include <stdio.h>
int i=2.4, j=4.1;
int main(void)
{
printf("i=%d j=%d \n",i,j);
return 0;
}
```

1. i=2 j=4
2. i=2.4 j=4.1
3. i=2.000000 j=4.000000
4. i=2.000000e+00 j=4.000000e+00
5. i=0 j=3

6. Что напечатает следующая программа

```
#include <stdio.h>
int i;
int main(void)
{
for (i=0;i<5;i++)
printf("i=%d\n",i);
return 0;
}
```

1. i=0
2. i=5
3. i=5.00
4. i=5.000000e+00
5. i=1

7. Что напечатает следующая программа

```
#include <stdio.h>
char i;
int main(void)
{
    i='A';
    switch(i)
    {
        case 'A':printf("A");break;
        case 'B':printf("B");break;
        case 'C':printf("C");break;
        default: printf("Not A,B,C");break;
    }
    return 0;
}
```

1. A
2. B
3. C
4. AB
5. Not A, B, C

8. Что напечатает следующая программа

```
#include <stdio.h>
char i;
int main(void)
{
    i='D';
    switch(i)
    {
        case 'A':printf("A");break;
        case 'B':printf("B");break;
        case 'C':printf("C");break;
        default: printf("Not A,B,C");break;
    }
    return 0;
}
```

1. A
2. B
3. C
4. D
5. Not A, B, C

9. Что напечатает следующая программа

```
#include <stdio.h>
int i;
int main(void)
{
    i=0;
    while(i<=10)
    {
        i++;
    }
    printf("i=%d",i);
    return 0;
}
```

1. i=0
2. i=1
3. i=10
4. i=11
5. i=5

10. Что напечатает следующая программа

```
#include <stdio.h>
int i;
int main(void)
{
    i=0;
    while(i>=10)
    {
        i++;
    }
    printf("i=%d",i);
    return 0;
}
```

1. i=0
2. i=1
3. i=10
4. i=11
5. i=5

11. Что напечатает следующая программа

```
#include <stdio.h>
int i;
int main(void)
{
    i=0;
    do {
        i++;
    } while(i>=10);
    printf("i=%d",i);
    return 0;
}
```

1. i=0
2. i=1
3. i=10
4. i=11
5. i=5

12. Что напечатает следующая программа

```
#include <stdio.h>
int a,b,max;
int main(void)
{
a=2;b=3;
if (a>b) max=a; else max=b;
printf("max=%d",max);
return 0;
}
```

1. max=0
2. max=2
3. max=3
4. max=5
5. max=1

13. Назовите тип переменной char в языке Си

1. целое
2. вещественное
3. символьное
4. вещественное с повышенной точностью
5. целое без знака

14. Назовите тип переменной int в языке Си

1. целое
2. вещественное
3. символьное
4. вещественное с повышенной точностью
5. целое без знака

15. Назовите тип переменной unsigned int в языке Си

1. целое
2. вещественное
3. символьное
4. вещественное с повышенной точностью
5. целое без знака

16. Назовите тип переменной float в языке Си

1. целое
2. вещественное
3. символьное
4. вещественное с повышенной точностью
5. целое без знака

17. Назовите тип переменной double float в языке Си

1. целое
2. вещественное
3. символьное
4. вещественное с повышенной точностью
5. вещественное с двойной точностью

18. Назовите тип переменной long double float в языке Си

1. целое
2. вещественное
3. символьное
4. вещественное с повышенной точностью
5. длинное вещественное с двойной точностью

19. Назовите тип переменной short int в языке Си

1. целое
2. короткое целое
3. символьное
4. вещественное с повышенной точностью
5. целое без знака

20. Назовите тип переменной bool в языке Си

1. целое
2. короткое целое
3. символьное
4. вещественное с повышенной точностью
5. целое без знака

21. Назовите назначение управляющих символов /n в языке Си

1. Переход к новой строке
2. Возврат каретки
3. Горизонтальная табуляция
4. Вертикальная табуляция
5. Переход к новой странице

22. Назовите назначение управляющих символов /r в языке Си

1. Переход к новой строке
2. Возврат каретки
3. Горизонтальная табуляция
4. Вертикальная табуляция
5. Переход к новой странице

23. Назовите назначение управляющих символов /t в языке Си

1. Переход к новой строке
2. Возврат каретки
3. Горизонтальная табуляция
4. Вертикальная табуляция
5. Переход к новой странице

24. Назовите назначение управляющих символов /v в языке Си

1. Переход к новой строке
2. Возврат каретки

3. Горизонтальная табуляция
4. Вертикальная табуляция
5. Переход к новой странице

25. Назовите назначение управляющих символов /f в языке Си

1. Переход к новой строке
2. Возврат каретки
3. Горизонтальная табуляция
4. Вертикальная табуляция
5. Переход к новой странице

26. Назовите тип, ожидаемый при вводе символ %d

1. Десятичное целое
2. Десятичное целое без знака
3. Символ
4. Символьная строка
5. Величина с плавающей точкой из мантиссы и порядка

27. Назовите тип, ожидаемый при вводе символ %u

1. Десятичное целое
2. Десятичное целое без знака
3. Символ
4. Символьная строка
5. Величина с плавающей точкой из мантиссы и порядка

28. Назовите тип, ожидаемый при вводе символ %c

1. Десятичное целое
2. Десятичное целое без знака
3. Символ
4. Символьная строка
5. Величина с плавающей точкой из мантиссы и порядка

29. Назовите тип, ожидаемый при вводе символ %s

1. Десятичное целое
2. Десятичное целое без знака
3. Символ
4. Символьная строка
5. Величина с плавающей точкой из мантиссы и порядка

30. Назовите тип, ожидаемый при вводе символ %f

1. Десятичное целое
2. Десятичное целое без знака
3. Символ
4. Символьная строка
5. Число с плавающей точкой

31. Какая схема отображает путь данных при решении задач и определяет этапы обработки, а также различные применяемые носители данных

1. Схема данных
2. Схема программы
3. Схема работы схемы
4. Схема взаимодействия программ

5. Схема ресурсов системы

32. Какая схема отображает последовательность операций в программе

1. Схема данных
2. Схема программы
3. Схема работы схемы
4. Схема взаимодействия программ
5. Схема ресурсов системы

33. Какая схема отображает управление операциями и поток данных в системе

1. Схема данных
2. Схема программы
3. Схема работы схемы
4. Схема взаимодействия программ
5. Схема ресурсов системы

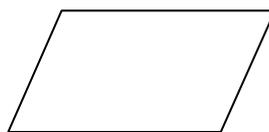
34. Какая схема отображает путь активации программ и взаимодействий с соответствующими данными

1. Схема данных
2. Схема программы
3. Схема работы схемы
4. Схема взаимодействия программ
5. Схема ресурсов системы

35. Какая схема отображает конфигурацию блоков данных и обрабатывающих блоков требуемых для решения задачи или набора задач.

1. Схема данных
2. Схема программы
3. Схема работы схемы
4. Схема взаимодействия программ
5. Схема ресурсов системы

36. Что отображает символ?



1. Данные, носитель данных не определен
2. Хранимые данные в виде, пригодном для обработки, носитель данных не определен
3. Данные, хранящиеся в оперативном запоминающем устройстве
4. Данные, хранящиеся в запоминающем устройстве с последовательным доступом
5. Данные, хранящиеся в запоминающем устройстве с прямым доступом

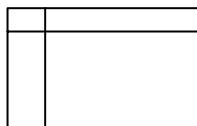
37. Что отображает символ?



1. Данные, носитель данных не определен
2. Хранимые данные в виде, пригодном для обработки, носитель данных не определен

3. Данные, хранящиеся в оперативном запоминающем устройстве
4. Данные, хранящиеся в запоминающем устройстве с последовательным доступом
5. Данные, хранящиеся в запоминающем устройстве с прямым доступом

38. Что отображает символ?



1. Данные, носитель данных не определен
2. Хранимые данные в виде, пригодном для обработки, носитель данных не определен
3. Данные, хранящиеся в оперативном запоминающем устройстве
4. Данные, хранящиеся в запоминающем устройстве с последовательным доступом
5. Данные, хранящиеся в запоминающем устройстве с прямым доступом

39. Что отображает символ?



1. Данные, носитель данных не определен
2. Хранимые данные в виде, пригодном для обработки, носитель данных не определен
3. Данные, хранящиеся в оперативном запоминающем устройстве
4. Данные, хранящиеся в запоминающем устройстве с последовательным доступом
5. Данные, хранящиеся в запоминающем устройстве с прямым доступом

40. Что отображает символ?



1. Данные, носитель данных не определен
2. Хранимые данные в виде, пригодном для обработки, носитель данных не определен
3. Данные, хранящиеся в оперативном запоминающем устройстве
4. Данные, хранящиеся в запоминающем устройстве с последовательным доступом
5. Данные, хранящиеся в запоминающем устройстве с прямым доступом

41. Что отображает символ?



1. Документ
2. Ручной ввод
3. Карту
4. Бумажную ленту
5. Дисплей

42. Что отображает символ?



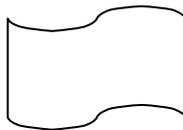
1. Документ
2. Ручной ввод
3. Карту
4. Бумажную ленту
5. Дисплей

43. Что отображает символ?



1. Документ
2. Ручной ввод
3. Карту
4. Бумажную ленту
5. Дисплей

44. Что отображает символ?



1. Документ
2. Ручной ввод
3. Карту
4. Бумажную ленту
5. Дисплей

45. Что отображает символ?



1. Документ
2. Ручной ввод
3. Карту
4. Бумажную ленту
5. Дисплей

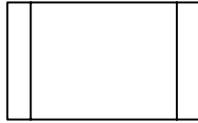
46. Что отображает символ?



1. Процесс
2. Предопределенный процесс
3. Ручная операция

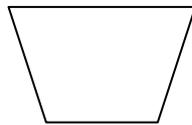
4. Подготовка
5. Решение

47. Что отображает символ?



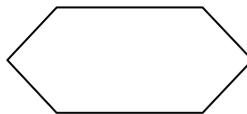
1. Процесс
2. Предопределенный процесс
3. Ручная операция
4. Подготовка
5. Решение

48. Что отображает символ?



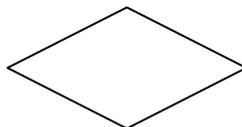
1. Процесс
2. Предопределенный процесс
3. Ручная операция
4. Подготовка
5. Решение

49. Что отображает символ?



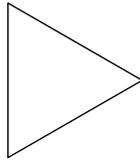
1. Процесс
2. Предопределенный процесс
3. Ручная операция
4. Подготовка
5. Решение

50. Что отображает символ?



1. Процесс
2. Предопределенный процесс
3. Ручная операция
4. Подготовка
5. Решение

51. Что отображает символ?



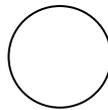
1. Передача управления
2. Канал связи
3. Соединитель
4. Терминатор
5. Комментарий

52. Что отображает символ?



1. Передача управления
2. Канал связи
3. Соединитель
4. Терминатор
5. Комментарий

53. Что отображает символ?



1. Передача управления
2. Канал связи
3. Соединитель
4. Терминатор
5. Комментарий

54. Что отображает символ?



1. Передача управления
2. Канал связи
3. Соединитель
4. Терминатор
5. Комментарий

55. Что отображает символ?



1. Передача управления
2. Канал связи
3. Соединитель
4. Терминатор
5. Комментарий

56. Итерация-это:

- 1) шаг выполнения операции
- 2) циклическая конструкция алгоритма
- 3) язык программирования
- 4) функция прерывания
- 5) ориентированная сеть

57. Дедуктивный принцип-это:

- 1) когда определяется связь между входными, выходными данными и процессами обработки
- 2) принцип построения модели от общего к частному
- 3) упрятывание информации
- 4) обработка элементов в алфавитном порядке
- 5) процесс нахождения в заданном множестве объекта

58. Индуктивный принцип – это:

- 1) когда определяется связь между входными, выходными данными и процессами обработки
- 2) принцип построения модели от частного к общему
- 3) упрятывание информации
- 4) обработка элементов в алфавитном порядке
- 5) процесс нахождения в заданном множестве объекта

59. Сортировка – это:

- 1) процесс нахождения в заданном множестве объекта
- 2) процесс перегруппировки заданного множества объектов в некотором порядке
- 3) установка индексов элементов в возрастающем порядке
- 4) обработка элементов в алфавитном порядке
- 5) суперпозиция двух алгоритмов

60. Тестирование программы – это:

- 1) оценивание ресурсов компьютера, на котором будет работать программа
- 2) перевод проекта в форму программы для конкретного компьютера
- 3) системный подход к построению алгоритма с использованием декомпозиции и синтеза
- 4) процесс исполнения программы с целью выявления ошибок
- 5) отслеживание логических ошибок

61. В языке Си лексема — это:

- 1) набор специальных символов и директив;
- 2) множество строк, определяющих состояние программы;

- 3) процедура, выполняющая определенные задания;
- 4) последовательности символов языка, разделяющиеся пробелами и другими не-графическими символами.
- 5) метка;

62. В языке Си указатель — это:

- 1) специальный значок, показывающий, что это динамическая переменная;
- 2) символическое представление адреса ячейки памяти;
- 3) символ, указывающий на что-либо;
- 4) метка;
- 5) набор специальных символов и директив.

63. В языке Си литерал — это:

- 1) переменная зарезервированного типа;
- 2) неизменяемый объект языка;
- 3) строка;
- 4) буква;
- 5) метка.

64. Комментарии заключаются в скобки:

- 1) { };
- 2) /* */;
- 3) [];
- 4) |% %|;
- 5) / /.

65. Идентификатор — это:

- 1) последовательность латинских букв, цифр и символа «_», начинающаяся с буквы или символа «_»;
- 2) неизменяемые объекты языка (константы);
- 3) последовательность латинских и русских букв;
- 4) способ кодирования, допустимые преобразования над значением данной переменной.
- 5) метка.

66. Фактический адрес в указателях — это:

- 1) строка;
- 2) указатель;
- 3) число;
- 4) буква.
- 5) последовательность латинских и русских букв;

67. Составной оператор — это:

- 1) последовательность операторов, заключенная в фигурные скобки {};
- 2) последовательность операторов, заключенная квадратные скобки [];
- 3) последовательность операторов, заключенная в операторные скобки begin ... end;
- 4) последовательность операторов, заключенная в круглые скобки ().
- 5) число.

68. Спецификация типа — это:

- 1) задание типа переменной;
- 2) список переменных;

- 3) перечисление всех переменных, которые использовались в программе;
- 4) список типов переменных, которые использовались в программе;
- 5) метка.

69. Логическое «и» обозначается:

- 1) =;
- 2) ||;
- 3) &;
- 4) &&;
- 5) !.

70. Логическое «не» обозначается:

- 1) !;
- 2) !!;
- 3) ||;
- 4) not;
- 5) &.

71. Битовая операция инверсии битов обозначается:

- 1) \~;
- 2) ~;
- 3) »;
- 4) « ;
- 5) ||.

72. Битовая операция исключающего «или» обозначается:

- 1) \~;
- 2) ~;
- 3) ||;
- 4) &&;
- 5) not.

73. Операция битового «и» обозначается:

- 1) \~;
- 2) ~;
- 3) ||;
- 4) &;
- 5) \.

74. Операция битового «или» обозначается:

- 1) \~;
- 2) ~;
- 3) |;
- 4) &;
- 5) not.

75. Текстовый поток — это:

- 1) логическое понятие, которое система может относить к чему угодно дисковых файлов до терминалов;
- 2) последовательность символов, которая организуется в строки, завершающиеся символами новой строки;

- 3) последовательность символов, которая организуется в списки слов, завершающиеся точкой с запятой;
- 4) текст программы;
- 5) метка.

76. Выражения — это:

- 1) конструкции, включающие константы (литералы), переменные, знаки операций, скобки для управления порядком выполнения операций, обращения к функциям;
- 2) основные строительные блоки программы; в языке Си указанием на наличие выражения служит символ «точка с запятой», стоящий в конце него;
- 3) набор символов и операций;
- 4) операторы, выполняющие определенные действия с переменными;
- 5) компактный способ записи оператора WHILE/DO.

77. Тернарное выражение — это:

- 1) компактный способ записи оператора с двумя операндами;
- 2) компактный способ записи оператора с тремя операндами;
- 3) выбор одного из нескольких вариантов;
- 4) выражение, описывающее действия логических связывающих операторов на переменные;
- 5) набор символов и операций.

78. Оператор-переключатель — это:

- 1) оператор для выбора одного из нескольких вариантов (SWITCH);
- 2) строка с меткой DEFAULT;
- 3) CASE;
- 4) BREAK.

79. Оператор цикла DO/WHILE является:

- 1) конструкцией цикла с предусловием;
- 2) конструкцией цикла с постусловием;
- 3) конструкцией цикла с выбором варианта;
- 4) конструкцией цикла с перебором значений параметра;
- 5) выражение, описывающее действия логических связывающих операторов на переменные.

80. Формальный аргумент — это:

- 1) конкретное значение, присвоенное этой переменной вызывающей программой;
- 2) переменная в вызываемой программе;
- 3) строка, которая пишется в скобках функции;
- 4) строка, которая пишется в скобках процедуры.
- 5) строка с меткой DEFAULT;