

Программа заданий конкурса-викторины
«Современная среда программирования»

для инженерных классов

1 тур

1.

Выберите правильную последовательность, соответствующую для интегрированной среда разработки:	
1.	Редактор кода, компиляция, отладка, редактор связей, выполнение программы
2.	Редактор кода, редактор связей, компиляция, выполнение программы, отладка
3.	Выполнение программы, редактор кода, компиляция, отладка, редактор связей,
4.	Редактор кода, отладка, редактор связей, выполнение программы, компиляция
5.	Редактор кода, компиляция, редактор связей, выполнение программы, отладка

2.

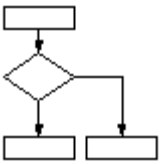
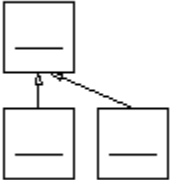
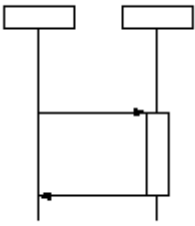
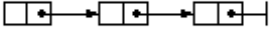
Выберите правильные пункты, включающие термины, которые характеризуют процесс отладки программы:	
1.	Редактор кода, компиляция, отладка, редактор связей, выполнение программы
2.	Редактор кода, редактор связей, компиляция, выполнение программы , отладка
3.	Точка остановки, пошаговое исполнение, компиляция, просмотр переменных, просмотр ассемблера
4.	Пошаговое исполнение, точка остановки, просмотр ассемблера, редактирование текста
5.	Точка остановки, пошаговое исполнение, просмотр памяти, просмотр переменных, просмотр ассемблера

3.

Выберите правильные пункты, определяющие термин «рефакторинг»	
1.	Преобразование программного кода в исполняемый
2.	Переименование имени переменной во всех файлах
3.	Автоматическое введение функций, переменных и структур

4.	Коррекция ошибок в исходном коде
5.	Модификация имён только в исходном коде программы

4.

Выберите правильные пункты, в которых представлена диаграмма последовательностей	
1.	
2.	
3.	
4.	

5.

Для чего используются команды «cl» или «gcc» в различных операционных системах	
1.	Вызов программного интерпретатора
2.	Вызов системных функций операционной системы
3.	Вызов препроцессоров языков программирования C/C++
4.	Вызов отладчика и дизассемблера языков программирования C/C++
5.	Вызов компиляторов языков программирования C/C++

6.

В каком виде сохраняется объектный файл программы

1.	Формат Microsoft Word
2.	Формат изображений jpeg и png
3.	Согласно заданной таблицы кодировки (Unicode, CP1251,...)
4.	Текстовый формат ASCII CP866
5.	Двоичный восьмибитный формат со всеми символами от 0 до 255

7.

Какие из перечисленных переменных имеют верные имена в большинстве языков программирования:	
1.	1A, 2Bc
2.	A_1, A_2
3.	#a, &c
4.	pA, Xa
5.	M a, B c

8.

Что характерно для компилятора любого языка программирования:	
1.	Возможность выбора значений переменных
2.	Возможность установки значений переменных препроцессора
3.	Возможность оптимизации кода по быстродействию
4.	Внесение исправлений в текст исходного кода
5.	Отладка программного кода

9.

В классе по сравнению со структурой данных отсутствуют:	
1.	Методы
2.	Значения с плавающей запятой
3.	Битовые поля
4.	Массивы
5.	Наследование

10.

Что характерно для отладки программного обеспечения:	
1.	Наличие точек останова при изменении данных
2.	Наличие точек останова в коде
3.	Модификация кода
4.	Пошаговое выполнение программы

5.	Перечень функций и классов
-----------	----------------------------

11.

Чем отличается объявление переменной от её определения:			
1.	Объёмом выделяемой памяти	2.	Отсутствием выделения памяти
3.	Разрядностью данных	4.	Адресом размещения в памяти
5.	Типом данных		

12.

Какие из особенностей образуют базу для любой интегрированной среды проектирования:	
1.	Наличие компилятора, редактора связей,
2.	Наличие файлового менеджера
3.	Наличие отладчика кода
4.	Наличие редактора рисунков
5.	Наличие текстового редактора

13.

Какие из этих языков являются компилируемыми:			
1.	C++	2.	Perl
3.	PHP	4.	Pascal
5.	Ruby		

14.

Какие ограничения существуют при использовании JTAG:	
1.	Количество строк исходного кода
2.	Количество точек остановок программы
3.	Количество загружаемых данных в ОЗУ
4.	Скорость передачи данных
5.	Количество функций ввода-вывода

15.

Что необходимо использовать для надёжной передачи различных данных в виде пакета	
1.	Тип данных
2.	Размер передаваемых данных
3.	Контрольная сумма
4.	Количество байт в единице данных
5.	Кодировка передаваемых данных

16.

Каким образом можно повысить быстродействие кода	
1.	Использовать макросы для ветвей алгоритмов
2.	Использовать подставляемые функции
3.	Использовать указатели на функции
4.	Использовать переключаемые условия «если-то» в виде switch-case
5.	Использовать рекурсивные функции

17.

Каким образом можно повысить эффективность кода	
1.	Использовать встроенные в компилятор расширения операций
2.	Использовать язык Ассемблера
3.	Применять выравнивание памяти
4.	Использовать макроподстановки
5.	Использовать побайтовые операции

18.

Какие из этих опций принадлежат компилятору	
1.	. Пути включаемых файлов
2.	Адреса размещения переменных в памяти
3.	Оптимизация кода по быстродействию
4.	Имя объектного файла
5.	Пути подключаемых библиотек

19.

Какие из этих опций принадлежат компоновщику	
--	--

1.	Имена библиотечных файлов
2.	Оптимизация по размеру кода
3.	Адреса размещения символов в памяти
4.	Имя исполняемого файла
5.	Пути подключаемых заголовочных файлов

20.

Какой из этих форматов представления чисел, обычно, имеет разрядность 32 бита			
1	. int	2.	unsigned short (word)
3.	char	4.	float
5.	double		

21.

Для чего необходим стек вызовов функций в отладчике:	
1.	Просмотр вложенности вызванных функций
2.	Сохранение результатов пошаговой отладки
3.	Взаимодействие с переменными функций
4.	Набор точек остановки внутри функции
5.	Детектирование перехода по условию

22.

Для чего необходим стек вызовов функций в отладчике:	
1.	Просмотр вложенности вызванных функций
2.	Сохранение результатов пошаговой отладки
3.	Взаимодействие с переменными функций
4.	Набор точек остановки внутри функции
5.	Детектирование перехода по условию

23.

Какие команды применяются при отладке:	
1.	Шаг с заходом
2.	Шаг с выходом
3.	Шаг двойной
4.	Шаг с обходом
5.	Шаг вниз

24.

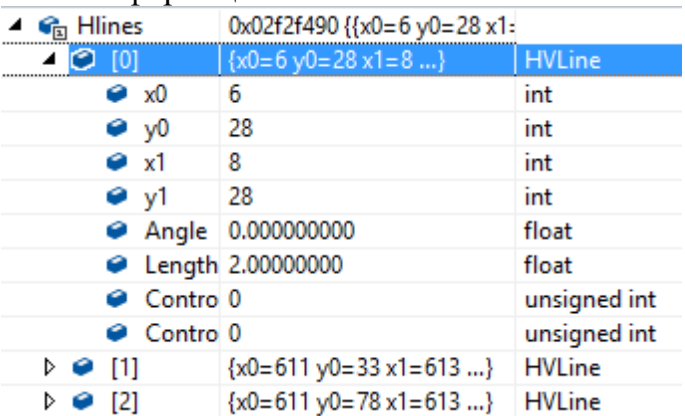
Что такое указатель:

1.	Ячейка памяти со значением адреса ячейки памяти
2.	Значение адреса следующей ячейки памяти
3.	Число, равное разнице адресов ячеек памяти
4.	Массив данных
5.	Номер регистра, хранящего адрес ячейки памяти

25.

Что такое константа:	
1.	Значение в постоянном запоминающем устройстве
2.	Значение, которое проверяется компилятором на изменение со стороны кода
3.	Значение, которое записано в памяти и проверяется на изменение программой
4.	Значение данных в файле с атрибутом «только для чтения»
5.	Элемент массива с одинаковыми значениями

26.

Дано окно отладочной информации:	
	
Выберите правильный ответ о том, какой тип данных используется для отладки	
1.	Массив структур с четырьмя целыми, двумя числами с плавающей запятой, двумя целыми беззнаковыми
2.	Список структур с четырьмя целыми, двумя числами с плавающей запятой, двумя целыми беззнаковыми
3.	Перечисление в памяти четырёх целых, двух чисел с плавающей запятой, двух целых беззнаковых значений
4.	Строковые значения “x0,y0,x1,y1...” с четырьмя целыми, двумя числами с плавающей запятой, двумя целыми беззнаковыми в текстовом виде
5.	Неструктурированные периодически повторяющиеся данные

27.

Что такое динамически подключаемая библиотека	
1.	Набор функций, которые размещаются в памяти и подготавливаются

	к вызову с использованием имени функции
2.	Набор данных, сформированных в виде сортированных значений
3.	Исходный код программы, который подключается командой препроцессора “include” в различных языках программирования
4.	Массив указателей на функции или виртуальные функции классов, организуемые динамически
5.	Каталог файлов, которые каждый раз сканирует программа на их наличие

28.

Выберите правильную последовательность действий в процессе формирования исполняемого файла и запуска программы на отладку	
1.	Редактирования текста, препроцессор, компилятор, редактор связей, исполняемый файл, отладка
2.	Препроцессор, редактирование текста, исполняемый файл, компилятор, отладка, редактор связей
3.	Редактирования текста, препроцессор, редактор связей, компилятор, исполняемый файл, отладка
4.	Редактирования текста, компилятор, препроцессор, редактор связей, исполняемый файл, отладка
5.	Редактор связей, редактор текста, препроцессор, компилятор, исполняемый файл, отладка

29.

При перемножении двух 16-ти битных значений получается:	
1.	два 16-ти битных значения
2.	одно 32-х битное значение
3.	одно 8-ми битное значение
4.	одно 17-ти битное значение
5.	два 32-х битных значения

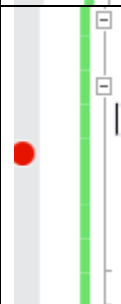
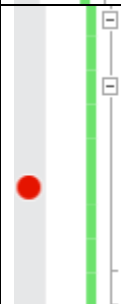
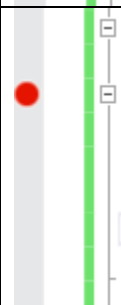
30.

Чем логическое «Или» отличается от двоичного «Или»:	
1.	результатом является значение равную нулю или отличное от него, интерпретируемое как «ложь» и «истина»
2.	результатом является значение равную нулю или единице, интерпретируемое как «ложь» и «истина»
3.	результатом является побитовая сумма значений, интерпретируемая как «ложь» и «истина»
4.	результатом является значение флага переноса, если он равен нулю, то выражение интерпретируется как «ложь» в противном случае - «истина»
5.	результатом является значение специального регистра, хранящего 0 или 1, интерпретируемое как «ложь» и «истина»

2 тур

1

Какие из этих таблиц расположения элементов в памяти соответствуют структуре данных «двусвязный список»		
Адрес	Значение	1
0	-	
10	5	
20	0	
30	40	
40	20	
50	10	
60	70	
Адрес	Значение	2
0	-	
10	5	
20	10	
30	15	
40	20	
50	25	
60	30	
Адрес	Значение	3
0	-	
10	20	
20	10	
30	30	
40	20	
50	40	
60	50	
Адрес	Значение	4
0	-	
10	0	
20	5	
30	20	
40	40	
50	10	
60	70	

Какой из фрагментов кода имеет правильно размещённую точку останова	
 <pre> for (int k = 0; k < PwthWCount; k++) { for (int j = 0; j <= PathW[k].PathIdx; j++) { float x0, y0, x1, y1, xm, ym; xm = USlines[PathW[k].Path[j]].x0; MiddlePointsArr[MiddlePointsCnt].xm = xm; } } </pre>	1
 <pre> for (int k = 0; k < PwthWCount; k++) { for (int j = 0; j <= PathW[k].PathIdx; j++) { float x0, y0, x1, y1, xm, ym; xm = USlines[PathW[k].Path[j]].x0; MiddlePointsArr[MiddlePointsCnt].xm = xm; } } </pre>	2
 <pre> for (int k = 0; k < PwthWCount; k++) { for (int j = 0; j <= PathW[k].PathIdx; j++) { float x0, y0, x1, y1, xm, ym; xm = USlines[PathW[k].Path[j]].x0; MiddlePointsArr[MiddlePointsCnt].xm = xm; } } </pre>	3
 <pre> for (int k = 0; k < PwthWCount; k++) { for (int j = 0; j <= PathW[k].PathIdx; j++) { float x0, y0, x1, y1, xm, ym; xm = USlines[PathW[k].Path[j]].x0; MiddlePointsArr[MiddlePointsCnt].xm = xm; } } </pre>	4

Какие из этих операций реализуют умножение на 4? ($y \ll x$ - арифметический сдвиг y влево на x разрядов)	
$B = a * 4$	1
$B = a \ll 1$	2
$B = a + a;$ $B = B + B;$	3

$B = a \ll 2$	4
---------------	----------

4

Что представлено на экране:	
<pre> 02659DD imul eax,dword ptr [PwthWCount (020BD484h)],34h 02659E4 mov ecx,4 02659E9 imul edx,ecx,0 02659EC mov ecx,dword ptr [ebp-0B48h] 02659F2 mov dword ptr PathW (020BD488h)[eax+edx],ecx </pre>	
Дизассемблированный код	1
Программа на языке ассемблера	2
Программа на языке «C»	3
Содержимое ячеек памяти, начиная с заданного адреса в виде системы команд процессора	4

5

Что такое «куча»?	
Структурированная область памяти, предназначенная для динамического выделения и высвобождения заданного количества памяти	1
Набор случайных данных в памяти	2
Указатель на произвольный массив данных, предоставляемый операционной системой	3
Выделяемые структурированные области памяти по заданным адресам	4

6

Имеется число с типом данных «знаковое слово» (short, word...). Какие из возможных вариантов граничных значений ему подходят:	
-32767...32768	1
-32768...32767	2
-1...128	3
-32768...32768	4

7

Что такое стек?	
Данные, организованные по принципу: «значение, поступившее первым, извлекается первым с начала массива»	1
Данные, организованные по принципу: «значение, поступившее первым, извлекается последним с начала массива»	2
Данные, организованные по принципу: «значение, поступившее первым в начало массива, извлекается первым с конца массива»	3
Данные, организованные по принципу: «значение, поступившее последним в начало массива, извлекается первым с конца массива»	4

8

В окне отладке имеется значение переменной 0xCAFE. Переведите в двоичный вид.	
0b1100101011111110	1
0b0100101011011110	2
0b0100101001011110	3
0b0100100011011100	4

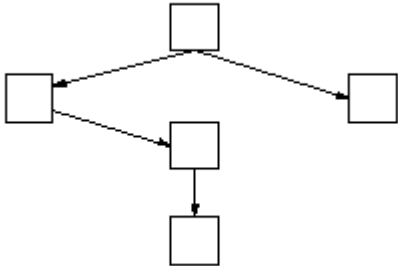
9

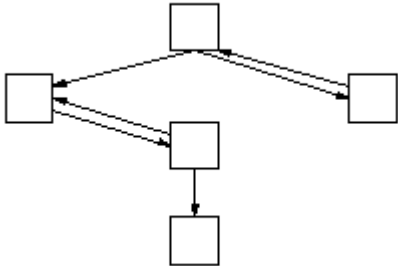
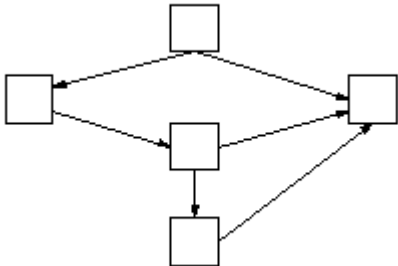
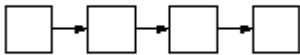
Чем глобальная переменная отличается от локальной Для глобальной:	
Память для глобальной переменной выделяется до запуска программы	1
Имеет доступ из всех исходных кодов программы	2
Память высвобождается до завершения программы	3
Выделяется регистровая ячейка памяти	4

10

Что такое профилирование	
Сбор данных о быстродействии программы во время её выполнения	1
Сбор данных о быстродействии программы во время компиляции	2
Программные элементы, составленные для определения времени выполнения	3
Код, сгенерированный компилятором для определения быстродействия	4

11

Какая из этих структур данных удовлетворяет циклическому графу	
	1

	2
	3
	4

12

Что такое массив указателей	
Ячейки памяти, располагаемые последовательно и содержащие непосредственные значения переменных	1
Ячейки памяти, располагаемые последовательно и содержащих адреса других ячеек памяти	2
Ячейки памяти, располагаемые в произвольном порядке и содержащих адреса других ячеек памяти	3
Ячейки памяти, располагаемые по заданным адресам	4

13

Что такое прерывание	
Вызов подпрограммы по алгоритму выполнения программы	1
Вызов подпрограммы по аппаратному сигналу	2
Вызов подпрограммы по специальной таблице указателей программой пользователя	3
Вызов подпрограммы по условному выражению	4

Для чего необходимо отключать оптимизацию кода при компиляции:	
Для просмотра значений глобальных переменных при отладке	1
Для просмотра значений стека при отладке	2
Для просмотра значений локальных переменных при отладке	3
Для просмотра макроподстановок при отладке	4

Что такое область видимости	
Область программного кода, в которой определяются локальные имена переменных	1
Область программного кода, в которой определяются глобальные имена переменных	2
Область программного кода, в которой определяются имена функций	3
Область программного кода, в которой определяются структуры данных	4

Что такое хэширование?	
Преобразование массива данных любой длины в определённую последовательность фиксированной длины	1
Контрольная сумма элементов массива	2
Преобразование массива данных любой длины в определённую последовательность произвольной разрядности	3
Применение операции логическое «исключающее ИЛИ» ко всем элементам массива	4

Что такое make-файл	
Файл, который содержит исходный код программы пользователя	1
Файл, который содержит последовательность построения проекта исходных кодов пользователя	2
Файл, который содержит сведения о размещении элементов в памяти, приводимые редактором связей	3
Исполняемый файл, который копирует и компилирует файлы исходного кода в папке проекта	4