**Вопросы по дисциплине ОП в ГУИМЦ,**

**выносимые на экзамен (2019/2020 уч.год).**

1. Понятие консольного проекта в системе программирования MS VS. Составляющие проекта. Понятие модуля. Русификация проекта.
2. Порядок обработки программы на языке СИ. Понятие модуля. Компилятор, редактор связей и исполнимый модуль. Отладчик. (Рисунок технологии обработки программ).
3. Структура программы на языке СИ. Основные и заголовочные модули проекта и библиотек. Понятие проекта. Его назначение и его преимущества.
4. Переменные, константы и выражения в языке СИ. Понятие переменной. Их назначение, типы, их инициализация и использование. Примеры
5. Операции и выражения в языке СИ. Знаки операций и их приоритеты. Бинарные и унарные операции. Условная операция. Примеры.
6. Математические вычисления в СИ. Библиотека математических функций. Понятие Оперативной памяти, ее назначение. Что и как в ней хранится. Рисунок.
7. Комментарии в СИ и их использование и разновидности. Примеры
8. Локальные и глобальные переменные. Понятие области видимости переменных. Доступ к переменным из другого модуля (extern). Примеры.
9. Модель выполнения программы. Ветвления и циклы, переходы. Расположение описания переменных в тексте программы. Пример.
10. Блок-схемы программ. Элементы блок-схем и правила оформления.
11. Описания, операторы и составные операторы. Операторы и операторные скобки в языке СИ. Примеры.
12. Оператор присваивания и выражение присваивания. Их применение и составляющие.
13. Вычисление значений переменных. Оператор присваивания. Особенности изменение значений переменной с использованием функций. Примеры.
14. Библиотеки в СИ и заголовочные файлы. Их использование в программах.
15. Библиотеки (назначение и состав): математическая, преобразований, ввода-вывода, стандартная, работа со строками и др.
16. Преобразование типов и библиотека преобразований строк. Cast – выражения.
17. Функции ввода и вывода в языке СИ. Ввод с клавиатуры и вывод на экран. Примеры. Уровни ввода и вывода.
18. Форматированный ввод вывод (функции printf scanf).
19. Отладка программ. Отладчики и их возможности. Графическая интерпретации отладки. Режимы, возможности и принципы отладки.
20. Операторы условного ветвления и вычисления. Метки и оператор безусловной передачи управления. Примеры.
21. Понятие цикла и его три основных составляющих. Блок-схема циклической программы. Разновидности операторов цикла.
22. Виды операторов цикла в языке СИ (for,do,while). Их особенности и примеры.
23. Специальные операторы управления и их назначение (break, continue,return).
24. Оператор переключатель (switch-case) его работа и блок-схема. Пример.
25. Массивы определение. Массивы и их типы. Инициализация массивов. Примеры.
26. Одномерные массивы. Размерность и размер массива. Особенности задания размерности массива в языке СИ. Элемент массива. Индексное выражение. Размер массива при инициализации массива. Примеры.
27. Двумерные и многомерные массивы. Их описание, использование и инициализация. Примеры.
28. Ввод вывод массивов.
29. Указатели: Понятие указателя и их использование. Назначение указателей. Операции именования и разыменования. Примеры и рисунок.
30. Инициализация и вычисление указателей. Массивы указателей. Указатели на массивы. Адресная арифметика и ее особенности.
31. Описание указателя на указатель и его использование. Примеры.
32. Описание составных типов переменных. Оператор typedef.
33. Динамическая память (ДП). Функции и операторы выделения и освобождения динамической памяти. Преимущества использования ДП. Библиотека ДП.
34. Динамические массивы. Создание одномерного динамического массива. Пример. Освобождение памяти выделенной под массив.
35. Строки в СИ. Их описание и использование. Инициализация строк. Основные функции для работы со строками. Понятие null-terminated строки. Примеры.
36. Библиотеки для работы со строками в СИ. Основные функции и их назначение.
37. Операции над строками. Динамические строки. Примеры.
38. Ввод вывод строк. Преобразование чисел в строку и обратно. Библиотека функций для преобразования.
39. Анализ символов (функции) и сравнение строк.
40. Аргументы командной строки и их вывод.
41. Массивы строк и массивы указателей на строки. Пример и рисунок.
42. Функции и процедуры в СИ. Параметры, передача параметров и возврат значений из функции. Формальные и фактические параметры. Теги в прототипах функции.
43. Понятие функции в СИ. Описание, прототип и вызов функций.
44. Стандартные библиотеки функций в СИ. Разновидности библиотек. Подключение библиотек функций в программу. Рекурсивные функции.
45. Основные функции и их назначение. Примеры. Проекты и функции. Указатели на функции. Передача параметров по значению и по ссылке (указателю). Изменение значений фактических параметров в функции.
46. Структуры данных. Шаблоны структуры данных и описание структурных переменных и их массивов Структурные переменные в структурах. Примеры. Поля структур и работа с ними через структурную переменную и через указатель.
47. Структуры данных. Массивы структур данных.
48. Сравнение массивов и структур.
49. Квалифицированная ссылка. Вложенные и локальные структуры. Пример.
50. Многоуровневая квалифицированная ссылка. Пример.
51. Создание динамической структуры. Работа с полями структуры с помощью имени структурной переменной и указателя.
52. Структуры и классы.
53. Функции и указатели, функции и массивы, функции и структуры (параметры и возврат).
54. Понятие алгоритма. Сортировка данных. Пузырьковая сортировка данных в массиве. Минимакс сортировка.
55. Поиск максимума и минимума в массиве.
56. Файлы. Понятие и характеристики файла. Основные операции с файлами.
57. Разновидности файлов. Их особенности. Просмотр файлов.
58. Файлы и файл менеджеры. Конец файла и его проверка.
59. Уровни работы с файлами. Работа с файлом целиком.
60. Стандартные потоки ввода вывода. Обработка ошибок.
61. Библиотеки для работы с файлами.
62. Открытие и закрытие файлов. Проверки при открытии файлов.
63. Файлы. Текстовые. Основные операции с файлами (fprintf, fscanf …). Примеры.
64. Работа с файлами построчно и посимвольно. Проверка конца файла.
65. Файлы. Двоичные. Основные операции (fread, fwrite … ). Примеры.
66. Понятие список. Структура элемент списка и список. Голова, хвост списка.
67. Навигация по спискам. Добавление и удаление в списках.
68. Списки. Разновидности. Однонаправленный список. Операции. Рисунок.
69. Списки. Разновидности. Двунаправленный список. Рисунок.
70. Объединения и перечисления в языке СИ.Назначение. Примеры.
71. Сравнение списков и массивов.
72. Основные документы на программное обеспечение, их назначение..
73. Техническое задание: назначение и разработка.
74. Программа и методика испытания, назначение и разработка.
75. Техническое описание: назначение и разработка
76. Вопросы по Домашнему заданию по дисциплине ОП.

### Примеры задач для экзамена по курcу Основы программирования.

**Задача 1**. По фрагменту текста функции построить блок-схему алгоритма и пояснить его работу.

int Summ5 (int \* mas , int k ,int \* psum)

{

// НУ цикла

int sum = 0 ; // Временная переменная для вычисления суммы

// Цикл

for (int i = 0 ; i < k ; i++ )

sum = sum + mas[i]; // тело цикла

\*psum = sum ; // возвращаемое значение через указатель

return sum; // возвращаемое значение из функции

};

**Ответ 1:**

Функция предназначена для вычисления суммы массива целого типа (**mas** – указатель на массив) размерностью **k** целых чисел. Результат передается из функции посредством указателя **psum** и возвратом из самой функции. Блок-схема программы имеет вид.



**Задача 2.** По фрагменту текста программы построить блок-схему алгоритма и пояснить его работу.

char StrMas[] = "Пример(ААА) строки (А) для подсчета пробелов и Символов А !!! ";

int NumA = 0;

int NumbSpace = 0;

printf ( "Исходная строка: %s\n", StrMas);

// Цикл подсчета

for (int i = 0 ; (unsigned)i <= strlen(StrMas); i++ )

{

if ( StrMas[i] =='А' ) NumA++;

if ( StrMas[i] ==' ' )

{ NumbSpace++; StrMas[i] ='\*' ;};

};

// Вывод результата

printf ( "Число символов \"А\" = %2d \nЧисло пробелов = %2d \n" ,NumA , NumbSpace );

//

**Ответ 2:**

В данном фрагменте содержится цикл обработки строки **StrMas**, в котором подсчитывается число заглавных букв “**А**” (счетчик **NumA**) и число пробелов (счетчик **NumbSpace**). В процессе подсчета пробелов они заменяются на символ звездочка “\*”. Блок-схема программы имеет вид (см. пример 1-го рейтинга).

