

Тестовые вопросы 2 (“Язык программирования C++”)
по дисциплине "Информационные технологии" для студентов 1-го курса АФ (поток 1АЕ),
специальность “Электромеханика”

1. Массив в информатике – это...
 - 1) некоторое множество мест в памяти компьютера;
 - 2) некоторое множество мест в памяти компьютера, называемых элементами массива, к которым можно обратиться по одному имени переменной;
 - 3) некоторое множество мест в памяти компьютера, называемых элементами массива, к каждому из которых можно обратиться по своему имени.
2. Чем однозначно определяется каждый элемент массива?
 - 1) Именем массива.
 - 2) Именем массива и его порядковым номером в массиве (индексом).
 - 3) Индексом элемента.
3. Какой индекс у 3-его элемента массива `int B[100]`?
 - 1) 3
 - 2) 2
 - 3) 0
4. Можно ли объявить массив следующим образом: `int C[6]= {2, 4, 7, 11, 12, 13}`?
 - 1) Да
 - 2) Нет
 - 3) В некоторых случаях
5. Как объявляется массив в языке C++?
 - 1) `int A[1..20]`
 - 2) `mas: A[1..20] int`
 - 3) `int A[20]`
6. Какое значение будет присвоено переменной `X` в программе на C++ оператором `X= m[13]`?
 - 1) 12-го элемента
 - 2) 13-го элемента
 - 3) 14-го элемента
7. Сколько ошибок сделано при записи оператора `if (A>7) B=A;` если обрабатывается массив `A`?
 - 1) 3
 - 2) 2
 - 3) 0
8. В программе на C++ при обработке массива все действия выполняются...
 - 1) над массивом в целом
 - 2) над элементами массива
 - 3) только над индексами элементов массива
9. Каким оператором можно ввести с клавиатуры `n` элементов массива `X`?
 - 1) `for (i=0; i<=n; i++) cin>>X[i]`
 - 2) `for (i=0; i< n; i++) cin>>X[i]`
 - 3) `for (i=1; i<=n; i++) cin>>X[i]`
10. Как в программе на C++ будет выведен на экран массив `A[k]` оператором `for(i=0; i<k; i++) cout<<A[i]<<endl;`?
 - 1) В виде строки
 - 2) В виде столбца
 - 3) В виде матрицы
11. Любой массив в языке C++...
 - 1) имеет имя
 - 2) не имеет имени
 - 3) имена имеют только элементы массива
12. Задан массив `X = {X1, X2 ..., XN}`. Каким оператором можно вывести этот массив на экран?
 - 1) `cout<<X;`
 - 2) `for (i=0; i<n; i++) cout<<X;`
 - 3) `for (i=0; i<n; i++) cout<<X[i];`
13. В программе на C++ объявлен массив `int B[10];` Это означает, что массив `B` содержит...
 - 1) 9 элементов
 - 2) 10 элементов
 - 3) 11 элементов
14. Возможна ли следующая инициализация массива `B` программе на C++: `int B[6]={2, 4, 7};`?
 - 1) Возможно
 - 2) Не возможно
 - 3) Зависит от мастерства программиста
15. В программе на C++ при обработке массива индекс элемента массива может быть задан...
 - 1) Значением
 - 2) Выражением
 - 3) Неважно как
16. Есть массивы `A` и `B` одинакового размера и их элементы имеют тип `int`. Возможно ли в языке C++ копирование массива `A` в `B` операцией `B = A`?
 - 1) Возможно
 - 2) Невозможно
 - 3) В некоторых случаях
17. При каком условии можно копировать массив `A` в массив `B`?
 - 1) Массивы `A` и `B` разного размера и их элементы имеют одинаковый тип
 - 2) Массивы `A` и `B` одинакового размера и их элементы имеют разный тип
 - 3) Массивы `A` и `B` одинакового размера и их элементы имеют одинаковый тип
18. Какой оператор копирует массив `A[N]` в массив `B[N]`?
 - 1) `for (i=0; i<N; i++) B[i]=A[i];`
 - 2) `for (i=0; i<K; i++) B[i]=A[i];`
 - 3) `for (i=0; i<N; i++) A[i]=B[i];`
19. Какой результат выполнения оператора `for (i=0; i<8; i++) cout<<"A["<<i<<"]="<<A[i]; (A=1...8)`?
 - 1) `A[0]=1 A[1]=2 A[2]=3 A[3]=4 A[4]=5 A[5]=6 A[6]=7 A[7]=8 A[8]=8`
 - 2) `A[1]=1 A[2]=2 A[3]=3 A[4]=4 A[5]=5 A[6]=6 A[7]=7 A[8]=8`
 - 3) `A[0]=1 A[1]=2 A[2]=3 A[3]=4 A[4]=5 A[5]=6 A[6]=7 A[7]=8`

20. Что нужно изменить в операторе `for(i=0; i<k; i++) cout<<A[i]<<endl;` чтобы массив `A[k]` был выведен на экран в виде строки?
- 1) Вместо `i` поставить `k`
 - 2) Убрать <<endl
 - 3) Этого нельзя сделать
21. Каким образом в программе на C++ объявляется двумерный массив?
- 1) `double A[0..2][0..3];`
 - 2) `double A[2, 3];`
 - 3) `double A[2][3];`
22. В памяти компьютера двумерный массив располагается...
- 1) непрерывно строками
 - 2) непрерывно столбцами
 - 3) в виде обычной матрицы
23. Возможно ли следующее объявление массивов в C++: `int i, j, k, m; int A[i][j], B[k][m];`?
- 1) Возможно
 - 2) Не возможно
 - 3) если `i>j`, а `k>m`
24. Как в программе на C++ будет выведен на экран двумерный массив `A[k][k]` оператором `for(i=0; i<k; i++) for(j=0; j<k; j++) cout<<A[i][j]<<endl;` ?
- 1) В виде строки
 - 2) В виде столбца
 - 3) В виде матрицы
25. Что нужно изменить в операторе `for(i=0; i<k; i++) for(j=0; j<k; j++) cout<<A[i][j]<<endl;` чтобы массив `A[k][k]` был выведен на экран в виде строки?
- 1) Вместо `i` подставить `k`
 - 2) Убрать <<endl
 - 3) Вместо `j` подставить `k`
26. Как будет выведен на экран двумерный массив в фрагменте программы на C++?
- ```
int i, j, k, m;
for(i=0; i<k; i++)
 for (j=0; j<m; j++)
 cout<<A[i][j]<<" ";
```
- 1) В виде строки
  - 2) В виде столбца
  - 3) В виде матрицы
27. Как будет выведен на экран двумерный массив в фрагменте программы на C++?
- ```
int i, j, k, m
for(i=0; i<k; i++)
{
    for ( j=0; j<m; j++)
        cout<<A[i][j];
    cout<<endl;
}
```
- 1) В виде строки
 - 2) В виде столбца
 - 3) В виде матрицы
28. Что будет выведено на экран в фрагменте программы на C++?
- ```
for(i=0; i<n; i++)
 for (j=0; j<n; j++)
 cout<<A[0][j];
```
- 1) Первая строка матрицы, повторенная n раз
  - 2) Первый столбец матрицы, повторенный n раз
  - 3) Полная матрица
29. Что нужно изменить в фрагменте программы
- ```
int i, j, k, m;
for(i=0; i<k; i++)
    for ( j=0; j<m; j++)
        cout<<A[i][j]<<" ";
```
- для вывода массива `A[k][m]` на экран в виде матрицы?
- 1) Вместо `i` подставить `k`
 - 2) Добавить оператор `cout<<endl;`
 - 3) Вместо `j` подставить `m`
30. Почему при работе с массивами в C++ целесообразно использовать оператор цикла `for`?
- 1) Т.к. оператор цикла `for` безошибочный
 - 2) Т.к. известен размер массива, т. е. число повторений цикла
 - 3) Т.к. оператор цикла `for` быстроедействующий
31. Какое действие выполняет оператор `for (i=0; i<N; i++) B[i][j]= A[i][j];`?
- 1) Выводит на экран матрицу `B[i][j]`
 - 2) Выводит на экран матрицу `A[i][j]`
 - 3) Копирует матрицу `A[i][j]` в матрицу `B[i][j]`
32. При обращении к функции реальные и формальные параметры должны совпадать...
- 1) по количеству и месту
 - 2) по количеству, типу и месту
 - 3) по количеству и типу
33. Обращение к функции осуществляется...
- 1) по заглавию функции
 - 2) по имени функции
 - 3) с помощью специального оператора
34. Функции могут быть...

- 1) только с параметрами
 - 2) только без параметров
 - 3) с параметрами и без параметров
35. В программе используются две независимые функции **F1** и **F2**. В каком порядке они должны быть описаны в разделе прототипов?
- 1) Сначала **F1**, а затем **F2**
 - 2) Сначала **F2**, а затем **F1**
 - 3) В любом порядке
36. В массиве **A** необходимо определить максимальный элемент и его номер. Возможно ли этот блок оформить функцией?
- 1) Возможно
 - 2) Возможно в отдельных случаях
 - 3) Невозможно
37. В программе описано две функции **F1** и **F2**. Возможно ли в них использовать одинаковые переменные?
- 1) Возможно
 - 2) Не возможно
 - 3) Возможно в отдельных случаях
38. Отдельный блок программы целесообразно оформить функцией, если в нем есть...
- 1) один результат
 - 2) два и более результата
 - 3) Все равно сколько результатов
39. В программе описана функция **int f(int x)**. Возможно ли так обратиться к этой функции: **r = f(n)/(f(m)*f(n-m));**?
- 1) Возможно
 - 2) Невозможно
 - 3) Возможно в отдельных случаях
40. Укажите правильный прототип функции **FF**, использующей в качестве аргумента целый массив **W[n]**.
- 1) **double FF(double B[], int n);**
 - 2) **double FF(int W[], int n);**
 - 3) **FF(double B[], int n);**
41. Укажите правильное обращение к функции **FF**, использующей в качестве аргумента целый массив **W[n]**.
- 1) **func (int W[], n);**
 - 2) **FF(W, n);**
 - 3) **func (int W[][]);**
42. Когда функция не возвращает никакого значения, можно ли ее использовать в выражениях языка C++?
- 1) можно
 - 2) нельзя
 - 3) можно, когда выражение принимает определенное значение
43. Параметры функции могут быть переменные, которые...
- 1) объявлены в функции
 - 2) объявлены вне функции
 - 3) все равно где
44. Укажите правильную запись заглавия функции:
- 1) **func(int x, v, float z)**
 - 2) **func(x, int v, float z)**
 - 3) **func(int x, int v, float z)**
45. Оператор **return**...
- 1) вызывает выход из функции
 - 2) используется для возвращения значения функции
 - 3) и первое, и второе
46. В теле функции оператор **return**...
- 1) может отсутствовать
 - 2) должен быть обязательно
 - 3) все зависит от назначения функции
47. Когда в теле функции отсутствует оператор **return**, выход из функции осуществляется...
- 1) когда выполнится последний оператор функции
 - 2) в данном случае возникнет зависание программы
 - 3) оператор **return** должен быть в теле функции обязательно
48. Если функция не имеет аргументов, то в прототипе такой функции необходимо...
- 1) ничего не писать в скобках функции
 - 2) в скобках записать служебное слово **int**
 - 3) в скобках записать служебное слово **void**
49. Локальная переменная – это переменная, которая объявлена...
- 1) в теле данной функции
 - 2) вне данной функции
 - 3) в качестве параметра в заглавии функции
50. В заглавии функции записываются...
- 1) формальные параметры
 - 2) фактические параметры
 - 3) глобальные переменные
51. При обращении к функции реальные параметры должны совпадать с формальными...
- 1) по месту

- 2) по месту и типу
3) по месту, типу и количеству
52. Что будет выведено на экран после выполнения программы `int A=300; cout<<&A;?`
- 1) Значение **A**, то есть 300
 - 2) Адрес ячеек, в которых записано значение A
 - 3) Сообщение об ошибке
53. Укажите правильное объявление указателя в C++.
- 1) ***double pA;**
 - 2) **double *pA;**
 - 3) **double pA*;**
54. Возможна ли следующая инициализация указателя: `char A="yes"; char *pA=&A;?`
- 1) Возможна
 - 2) Невозможна
 - 3) Такой конструкции в C++ нет
55. Что означает оператор `double *pA=&A;?`
- 1) Инициализация переменной **A**
 - 2) Объявление указателя **pA**
 - 3) Инициализация указателя **pA** и присваивание ему адреса переменной **A**
56. Что такое указатель в языке C++?
- 1) Адрес некоторой переменной или функции
 - 2) Переменная, содержащая адрес некоторой переменной или функции
 - 3) Стандартная функция
57. Как объявляется указатель с именем **pA** в языке C++?
- 1) **int *pA**
 - 2) **int «pA**
 - 3) **int &A**
58. Как обозначается операция взятия адреса переменной **A** в языке C++?
- 1) ***pA**
 - 2) **«pA**
 - 3) **&A**
59. Как обозначается операция разыменования указателя **pA** в языке C++?
- 1) ***pA**
 - 2) **«pA**
 - 3) **&pA**
60. Что означает операция разыменования указателя в языке C++?
- 1) Получение адреса переменной в указателе
 - 2) Получение значения переменной, записанной по адресу, который находится в указателе
 - 3) Запись переменной по адресу, который находится в указателе
61. Какое значение примет **Y** в программе `double X=10.1; double *pA; pA=&X; Y=*pA;`
- 1) **Y=10.1**
 - 2) В переменную **Y** запишется адрес, по которому находится значение **X**
 - 3) Будет выдано сообщение об ошибке с указанием на последнюю строку
62. Можно ли в языке C++ выполнять арифметические операции над указателями?
- 1) Можно
 - 2) Нельзя
 - 3) Можно выполнять только операцию присваивания
63. Что будет выведено на экран дисплея после выполнения программы `int *pA; for (i=0; i<100; i++) cout<<*(pA+i)<<" ";?`
- 1) Значения элементов какого-то массива
 - 2) Значения указателей
 - 3) Такую конструкцию в языке C++ использовать нельзя
64. Укажите на возможность такого объявления указателя `int **ppA;`
- 1) Возможно
 - 2) Невозможно
 - 3) Все зависит от содержания программы
65. На сколько байтов изменится значение **pC** в программе `int C[20]; int *pC=&C; pC++;?`
- 1) Не изменится
 - 2) Увеличится на 4 байта
 - 3) Уменьшится на 4 байта

Составитель: доц. Симбирский Г. Д.