Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра систем сбора и обработки данных

Паспорт экзамена

по дисциплине «Информатика», 1 семестр

1. Методика оценки

Экзамен проводится в письменной форме, по билетам. Билет состоит из 2 вопросов и формируется по следующему правилу:

- первый вопрос выбирается из диапазона вопросов 1-24;
- второй вопрос из диапазона вопросов 25-55.

Таким образом, проверяется уровень сформированности компетенций и соотнесенных с ними индикаторов, закрепленных за дисциплиной.

На экзамене преподаватель вправе задавать студенту уточняющие и дополнительные вопросы из общего перечня (п. 4).

Форма экзаменационного билета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____
к экзамену по дисциплине «Информатика»

1.Типы данных. Переменные. Байтовые, целые, числа с плавающей запятой. Структура этих типов.
2. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: void f(int *A, int *B, int n, int m);
Параметры: A - указатель исходной матрицы
B - указатель транспонированной матрицы
n – количество строк исходной матрицы
m – количество столбцов исходной матрицы
Назначение: Нахождение транспонированной матрицы.
Транспонирование заключается в обмене местами строк и столбцов матрицы. Элемент, расположенный в i-той строке и j-том столбце помещается в j-тую строку и i-тый столбец.

Утверждаю: зав. кафедрой ____ _____ должность, ФИО
(подпись)
(дата)

2. Уровни освоения компетенций и критерии оценки

Ответ на экзаменационный билет засчитывается на **продвинутом** уровне, если студент при ответе на вопросы проводит сравнительный комплексный анализ материала, выявляет проблемы, предлагает механизмы их решения, представляет количественные характеристики определенных процессов, приводит конкретные примеры, не допускает ошибок. Компетенции и соотнесенные с ними индикаторы, закрепленные за дисциплиной, сформированы в полном объеме. Оценка составляет *от* 36 *до* 40 *баллов.*

Ответ на экзаменационный билет засчитывается на **базовом** уровне, если студент при ответе на вопросы формулирует основные понятия, дает характеристику процессов, явлений, проводит анализ причин, условий, способен представить количественные и качественные характеристики процессов. Компетенции и соотнесенные с ними индикаторы, закрепленные за дисциплиной, содержат несущественные пробелы и сформированы на базовом уровне. Оценка составляет *от 30 до 35 баллов.*

Ответ на экзаменационный билет засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, может показать причинно-следственные связи явлений. Компетенции и соотнесенные с ними индикаторы, закрепленные за дисциплиной, содержат пробелы и сформированы на пороговом уровне. Оценка составляет *от* 20 *до 29 баллов.*

Ответ на экзаменационный билет считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не способен показать причинно-следственные связи явлений. Компетенции и соотнесенные с ними индикаторы, закрепленные за дисциплиной, не сформированы. Оценка составляет *менее* 20 *баллов.*

3. Шкала оценки

Экзамен считается сданным, если сумма баллов по всем заданиям билета составляет от 20 до 40 баллов включительно. Сумма менее 20 баллов признается неудовлетворительным результатом промежуточной аттестации по дисциплине (модулю).

В общей оценке по дисциплине экзаменационные баллы учитываются в соответствии с правилами балльно-рейтинговой системы, установленными в НГТУ.

4. Вопросы к экзамену по дисциплине «Информатика»

1. Системы счисления (Позиционные и непозиционные). Алгоритмы перевода из одной системы в другую. (Целые и дробные числа). Обратный и дополнительный код.
2. Типы данных. Переменные. Байтовые, целые, числа с плавающей запятой. Структура этих типов.
3. Операции и выражения. Перечисляемые константы. Преобразования типов (явные, неявные). Приоритеты и порядок выполнения.
4. Операторы языка. Составной оператор. Управляющие конструкции языка.
5. Условный оператор (if … else …). Переключатель (switch).
6. Операторы цикла (while , do while, for)
7. Операторы передачи управления (break , continue , goto , return)
8. Функции в С. Формальные и фактические параметры. Механизм передачи параметров. Возвращаемые значения. Использование указателей в качестве аргументов функции. Предварительное описание функций.
9. Адреса и указатели. Операции получения адреса и косвенная адресация. Отождествление массивов и указателей. Адресная арифметика.
10. Указатели на массивы. Массивы указателей и многомерные массивы. Динамическое

выделение памяти под массивы. Инициализация указателей.
11. Общая структура программы. Время существования и видимость переменных. Классы Памяти.
12. Структуры, объединения, перечисления.
13. Рекурсивные вызовы функций. Аргументы командной строки.
14. Внешние объекты программы (extern). Статические переменные и функции (static). Регистровые переменные (register).
15. Препроцессор языка Си. (#define #undef #include #if, #ifdef, #ifndef, #else, #elif и #endif)
16. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: int f(char *a, char * b);
Параметры: a - указатель исходной строки; b - указатель результирующей строки
Назначение: отсортировать символы в строке в алфавитном порядке. Алгоритм: поиск минимального в исходной строке, запись в выходную строку.
17. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: void f(char *a);
Параметры: a - указатель исходной строки
Назначение: Переставить в строке символы попарно слева и справа
(1..n, 2..n-1 и т.д.);
18. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: void f(char *a);
Параметры: a - указатель исходной строки
Назначение: Отсортировать символы в строке. Алгоритм: последовательно сравниваются пары соседних элементов xk и xk+1 (k=0, . . . ,n-2), если xk > xk+1 , то они переставляются.
19. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: void f(char *a);
Параметры: a - указатель исходной строки
Назначение: Вывести на экран в первой строке нечетные символы, во второй четные символы исходной строки.
20. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( ).
Прототип: void f(char *a);
Параметры: a - указатель исходной строки
Назначение: Вывести на экран сначала цифры, затем буквы, затем остальные символы исходной строки.
21. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( ).
Прототип: void f(char *a);
Параметры: a - указатель исходной строки
Назначение: Для каждого символа исходной строки найти число вхождений его в строке и напечатать в столбец (символ, частота вхождения).
22. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: void f(char *a , char * b);
Параметры: a - указатель исходной строки
b - указатель результирующей строки
Назначение: Исключить из исходной строки все пробелы и записать в результирующую строку.

23. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: int f(int *mat, int n, int m, int &min, int &max);
Параметры: mat - указатель исходной матрицы
n –количество строк
m –количество столбцов
min – минимальное значение
max – максимальное значение
Назначение: Вычисление максимального и минимального значений элементов матрицы
24. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: int f(int *mat, int n, int m, int & A, int & B);
Параметры: mat - указатель исходной матрицы
n –количество строк
m –количество столбцов
A – число элементов > 0
B– число элементов < 0
Назначение: Вычисление числа элементов больше нуля и меньше нуля
25. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: int f(int *mat, int n);
Параметры: mat - указатель квадратной матрицы
n –количество строк, столбцов
m –количество столбцов
Назначение: Заполнение матрицы последовательностью чисел от 0 до n*n, расположенными по спирали
1 2 3 4 5 6
20 21 22 23 24 7
19 32 33 34 25 8
18 31 36 35 26 9
17 30 29 28 27 10
16 15 14 13 12 11
26. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: void f(int *A, int *B, int n, int m);
Параметры: A - указатель исходной матрицы
B - указатель транспонированной матрицы
n – количество строк исходной матрицы
m – количество столбцов исходной матрицы
Назначение: Нахождение транспонированной матрицы.
Транспонирование заключается в обмене местами строк и столбцов матрицы. Элемент, расположенный в i-той строке и j-том столбце помещается в j-тую строку и i-тый столбец.
27. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: void f(int *A, int *X, int *Y, int n, int m);
Параметры: A - указатель матрицы
X - указатель вектора (одномерного массива)
Y – указатель результирующго вектора
n – количество строк матрицы
m – количество столбцов матрицы
Назначение: вычисление произведения матрицы на вектор.

В результате умножения матрицы A[m][n] на вектор X[n] получается вектор Y[m], компоненты которого вычисляются по формуле: y[i]=SUM (a[i][j]*x[j] , (i = 0, 2, …, m-1)
28. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: void f(int *A, int *B, int *C, int n, int m, int l);
Параметры: A - указатель первой матрицы
B - указатель второй матрицы
C - результирующей матрицы
n – количество строк первой матрицы
m – количество столбцов первой матрицы, количество строк
второй матрицы
l – количество столбцов второй матрицы
Назначение: вычисление произведения 2-х матриц (строка на столбец).
В результате умножения двух матриц A[m][n] и B[n][l] получается матрица C[m][l], элементы которой находятся по следующей форм уле:C[i][j]= SUM a[i][j]*b[k][j]
29. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: int f(int *A, int m);
Параметры: A - указатель квадратной матрицы
m – количество строк, столбцов матрицы
Назначение: вычисление среднего арифметического элементов, лежащих ниже диагонали
30. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: int f(int *a, int n);
Параметры: a - указатель исходного массива, n - размер исходного массива.
Назначение: в исходном массиве все элементы, встречающиеся больше одного раза заменить на нули.
31. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: int f(int *a, int n);
Параметры: a - указатель исходного массива, n - размер исходного массива.
Назначение: Отсортировать элементы массива по убыванию методом пузырька
Алгоритм : последовательно сравниваются пары соседних элементов xk и xk+1 (k=0, n-2), если xk < xk+1 , то они переставляются.
32. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: int f(int *a, int n);
Параметры: a - указатель исходного массива, n - размер исходного массива.
Назначение: Отсортировать элементы массива по следующему алгоритму: отыскивается максимальный элемент и переносится в конец массива, затем метод переносится
ко всем элементам, кроме последнего и т.д.
33. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: int f(int M, int N);
Параметры: M,N - целые числа
Назначение: Нахождение наибольшего общего делителя двух чисел N и M по рекуррентному соотношению
Nk = Nk-2 - INT(Nk-2 / Nk-1) N k-1 k=2,3 ...
N0 = max(|N|, |M|) N1 = min(|N|, |M|)
Если Nk = 0 => НОД = N k-1
(N=23345 M=9135 => 1015 N=238 M=347 => 34) 6
34. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям,

вызвать ее в программе main( )
2.1. Прототип: int f(int *a, int n);
2.2. Параметры: a - указатель исходного массива, n - размер исходного массива.
2.3. Назначение: в исходном массиве удаляются все элементы, значения которых меньше, чем количество их повторов в массиве. Порядок следования остальных
элементов массива не изменяется.
2.4. Возвращаемое значение: количество оставшихся элементов.
2.5. Пример использования:

```
int a[] = { 1, 2, 3, 4, 2, 4, 6, 2 ,4, 7, 1, 1, 5 };
int n;
printf( "n = %d\n", n = f( a, sizeof(a)/sizeof(int)));
for( int i = 0 ; i < n ; printf( "%d ", a[i++]));
// вывод на экран:
//
// n = 7
// 3 4 4 6 4 7 5
```

35. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
2.1. Прототип: int f(int* a, int n, int k);
2.2. Параметры: a - указатель исходного массива, n - размер массива, k - размер сдвига.
2.3. **Назначение:** исходный массив преобразуется в результате циклического сдвига на указанное число позиций. "Циклически" означает, что при выходе за верхнюю
границу массива запись производится в позиции 0, 1 и т.д., а при выходе за нижнюю - в позиции n-1, n-2 и т.д.
2.4. Возвращаемое значение: свой первый параметр.
2.5. Пример использования:

```
int i, x[] = { 3, 6, 4, 1, 8, 2, 5, 7, 0 };
f( x, sizeof(x)/sizeof(int), -3 );
for( i = 0 ; i < sizeof(x)/sizeof(int) ; ++i ) printf( "%d ", x[i] );
f( x, sizeof(x)/sizeof(int), 2 ); printf( "\n" );
for( i = 0 ; i < sizeof(x)/sizeof(int) ; ++i ) printf( "%d ", x[i] );
// вывод на экран:
//
// 1 8 2 5 7 0 3 6 4
// 6 4 1 8 2 5 7 0 3
//
```

36. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
2.1. Прототип: int f(char *a, float *b);
2.2. Параметры: a - указатель исходной строки, b - указатель буфера.
2.3. **Назначение:** в исходной строке ищутся и распечатываются в столбик все последовательности литер, представляющие записи целых чисел или действительных чисел в формате <целая часть>.<дробная часть>. Сами числа в формате float последовательно записываются в буфер.
2.4. Возвращаемое значение: количество найденных и записанных чисел.
2.5. Пример использования:

```
char s[] = "12.45, 678.99, -34 0.0004 ";
float B[100];
printf( "\nЗаписано чисел: %d", f( s, B ));
// вывод на экран:
//
// 12.450000
```

```
// 678.99000
// -34.000000
// 0.000400
// Записано чисел: 4
//
```
37. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
2.1. Прототип: int* f(int* a, int n);
2.2. Параметры: a - указатель исходного массива, n - размер массива.
2.3. **Назначение:** исходный массив преобразуется по следующему алгоритму: все нечетные элементы массива группируются в порядке возрастания в начале массива, а четные в порядке убывания в конце.
2.4. Возвращаемое значение: указатель первого четного элемента в преобразованном массиве.
2.5. Пример использования:
```
int x[] = { 3, 6, 4, 8, 2, 5, 7, 0 };
int *p = f( x, sizeof(x)/sizeof(int));
printf( "n = %d\n", p-x );
for( int i = 0 ; i < sizeof(x)/sizeof(int) ; ++i ) printf( "%d ", x[i] );
// вывод на экран:
//
// n = 3
// 3 5 7 8 6 4 2 0
//
```
38. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
2.1. Прототип: char* f(char *v1, char *v2, char *r, int n);
2.2. Параметры: v1, v2 - указатели исходных строк, r - указатель результирующей строки, n - размер результирующей строки.
2.3. Назначение: исходные строки являются символьным представлением натуральных чисел произвольной длины. Необходимо получить в результирующей строке такое
же представление суммы этих чисел. Не умещающиеся в результирующей строке цифры пропадают. Остающиеся свободные позиции заполняются кодом нуля.
2.4. Возвращаемое значение: указатель литеры, являющейся представлением первой значащей цифры результата.
2.5. Пример использования:
```
char a[] = "45789", b[] = "1111111", c[20];
printf( "\n%s\n%s", f( a, b, c, sizeof c ), c );
//
// вывод на экран:
//
// 1156900
// 0000000000001156900
//
```
39. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: double f(int n);
Назначение: y = 1! + 2! + 3! + 4! + ... +n!
40. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( ) 8
Прототип: double f(double x, double e);
Найти функцию с точностью e , т.е. пока |fn+1-fn| < e . Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине

определяется эта функция.
ex = 1 + x +x2/2! + x3/3! + ...
41. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: double f(double x, double e);
Найти функцию с точностью e , т.е. пока |fn+1-fn| < e . Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.
sin(x) = x - x3/3! + x5/5! + ...
42. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: double f(double x, double e);
Найти функцию с точностью e , т.е. пока |fn+1-fn| < e . Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.
cos(x) = 1 - x2/2! + x4/4! ...
43. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: double f(double x, double e);
Найти функцию с точностью e , т.е. пока |fn+1-fn| < e . Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.
ln(1+x) = x - x2/2 + x3/3 - ... |x| < 1
44. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: double f(double x, double e);
Найти функцию с точностью e , т.е. пока |fn+1-fn| < e . Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.
cos(x) = 1 - x2/2! + x4/4! ...
45. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: double f(double x, double e);
Найти функцию с точностью e , т.е. пока |fn+1-fn| < e . Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.
x
tg(x) = ----------
x2
1 - ----------
x2
3 - ----------
x2
5 - --------
7 -
46. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: double f(double e);
Найти функцию с точностью e , т.е. пока |fn+1-fn| < e . Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.
1

sqrt(2)=1 + ----------
1
2 + ----------
1
2 + ----------
1
2 + -------
2 + ...
47. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: double f(double e);
Найти функцию с точностью e , т.е. пока |fn+1-fn| < e . Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.
2
e =2 + ----------
3
2 + ----------
4
3 + ---------
5
4 + -------
5 + ...
48. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: double f(double x, double e);
Найти функцию с точностью e , т.е. пока |fn+1-fn| < e . Определить абсолютную разность между табличной функцией и суммой ряда. Определить с какой точностью в машине определяется эта функция.
49. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: void f(int n);
Печать римских цифр
1. 1- I 10-X
2. 2-II 50-L
3. 3-III 100-C
4. 4-IV 500-D
5. 5-V 1000-M
6. 6-VI
Соблюдая следующие правила:
7. Если младшая цифра стоит перед старшей она вычитается IV-4
8. Если младшая цифра стоит после старшей - складывается VI -5
9. Число представляется минимальным количеством знаков III-3, 4-IV, а не IIII.
50. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: int f(int n, int k);
Напечатать подряд все натуральные числа до n - 012345678910111213 Найти к-ую цифру последовательности
51. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям
Найти все четырехзначные числа Армстронга. Натуральное число из n знаков является числом Армстронга, если сумма его цифр, возведенная в n-ю степень равна самому
числу. 153=13+53+33

52. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям
Вставить между цифрами 1,2,3,4...9 знак одной из операций +-*/, так, чтобы результат 8-опреаций равнялся 100. Имеется 8 решений. Распечатать их со знаками.
1*2+3*4*5-6+7+8-9=10
53. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям
Совершенными числами называются числа, равные сумме своих делителей (6=1+2+3). Найти все совершенные числа от 0 до 10000 и распечатать их вместе со своими делителями.
54. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: void f1(int n, *char a); // a – массив, состоящий из 0 или 1. В main( ) массив распечатать
Перевести из десятичной системы в двоичную целое число
327 : 2 = 163 остаток 1
163 : 2 = 81 1
81 : 2 = 40 1
40 : 2 = 20 0
20 : 2 = 10 0
10 : 2 = 5 0
5 : 2 = 2 1
2 : 2 = 1 0
1 : 2 = 0 1
32710 = 1010001112
55. Написать на языке "C" или "C++" функцию, удовлетворяющую следующим требованиям, вызвать ее в программе main( )
Прототип: void f1(double b, *char a); // a – массив, состоящий из 0 или 1. В main( ) массив распечатать
Перевести из десятичной системы в двоичную дробное
0.125 x 2 = 0.250 = 0 + 0.250
0.250 x 2 = 0.5 = 0 + 0.5
0.500 x 2 = 1.000 = 1 + 0.00
0.12510 =0.0012_