Требования к оформлению отчетов по лабораторным работам.

Отче оформляется по каждой лабораторной работе отдельным документом и предусматривает наличие следующих разделов:

1. Титульный лист (с указанием кафедры, вида работы, ФИО студента и преподавателя, учебной группы и факультета).
2. Задание на лабораторную работу.
3. Цели и задачи лабораторной работы.
4. Теоретические основы: здесь необходимо привести описание используемых впервые (в контексте данной ЛР) функций, технологий, приемов, особенности синтаксиса, входных/выходных параметров, и т.п.
5. Листинг программы, снабженный комментариями.
6. Блок-схемы функций: для каждой функции составляется отдельная блок-схема.

Отчеты печатаются на листах А4, строго с одной стороны листа. Допустимо рисование блок-схем от руки, как и написание текста отчета, при этом выставляется строгое требование разборчивости почерка и рисунков. Основные блоки блок-схем можно посмотреть, например, здесь https://pro-prof.com/archives/1462. Оформление блок-схем можно осуществлять в различных редакторах. При этом следует учитывать, что количество дуг, сопряженных с каждым углом многоугольных блоков циклов и инструкций for, while, if, и др. – не более одной. Все дуги должны быть строго ортогональны друг другу, перенос на другую страницу части блок-схемы сопровождать ссылками (по дугам), преимущественное направление циклических траекторий – по часовой стрелке.

Шрифт 12, поля стандартные (левое – 2 см, остальные – по 1.5 см), междустрочный интервал – одинарный или полуторный, абзацный отступ 1.25 см. Размер шрифта на картинках не менее 10го.

Лабораторная работа № 1: Ознакомление со средой программирования

Цель: знакомство и получение первичных навыков программирования в среде VisualStudioC++ / CodeBlocks / NetBeansC++ / DevC++ / CLion /…
Задание: реализовать программы для решения 10 различных задач. Предусмотреть использование:
- циклов while, for, do…while
- работу с массивами
- обработку строковой и символьной информации
- математических алгоритмов (поиск НОД, НОК, и т.п.)
- использование собственных функций, возвращающих типы int, void, char*, и т.п.
- работу с указателями
- форматированный вывод на экран (например, вывод таблицы Пифагора вместе с «шапкой»)
- генератор случайных чисел
- инструкции case, if
- работу с файлами (чтение/запись/добавление)
- реализацию алгоритмов для решения прикладных задач (игры – быки-коровы, крестики-нолики, математические формулы и алгоритмы)
- работа в различных системах счисления: перевод чисел, стандартный ввод/вывод чисел в различных системах счисления
- и т.д.

При вводе/выводе информации на экран и в файл НЕ использовать потоковые объекты. Вся работа должна быть реализована через функции printf/scanf.

Лабораторная работа №2: Сортировка массивов с визуализацией

Цель: изучить основные подходы к сортировке массивов, а так же способы визуализации процесса сортировки.
Задачи:
- изучить методы сортировки массивов;
- освоить понятия устойчивых и неустойчивых сортировок;
- изучить возможность визуализации процессов, используя выделение цветом и форматированный вывод, покадровый вывод (задержку экрана и очистку консоли);

Задание: Реализовать 3 метода сортировки массива с визуализацией.
Размерность массива может быть задана пользователем или константно, в диапазоне от 10 до 15 элементов. Значение элементов массива разыграть случайным образом, предусмотреть возможность корректировки пользователем предельных значений (границ диапазона).
Визуализация процесса сортировки должна отображать цветом или другим способом (кроме выделения скобками и т.п.) все сравнения и все перемещения элементов массива. При визуализации процесса сортировки можно использовать стандартные функции:
- задержка экрана;
- выделение символа и/или подложки цветом;
- очистка консоли;
- сдвиг курсора;
- и т.д.

При написании кода рекомендуется использовать стандартные функции ввода/вывода языка C – printf, scanf. Предусмотреть настройку ширины поля вывода для элементов массива.

Лабораторная работа №3: Битовые операции

Цель: изучить основные операции при работе с битовым представлением чисел.

Задачи:

1. Изучить принципы построения прямого, обратного, дополнительного кодов целых чисел;
2. Изучить способы хранения вещественных чисел в памяти компьютера;
3. Изучить алгоритм "Решето Эратосфена".

Задание:

1. Для введенного пользователем числа типа short предусмотреть возможность установки выбранного пользователем бита в 1, сброса выбранного пользователем бита в 0, изменения выбранного пользователем бита, опроса выбранного пользователем бита. Для реализации вышеуказанных функций использовать битовые операции.
2. Для чисел типа int, short, long long (и других целочисленных) вывести на экран прямой, обратный, дополнительный коды и внутреннее представление в памяти компьютера (dump). Исходное значение вводится пользователем или задается программистом в тексте программы. Предусмотреть возможность обработки как положительных, так и отрицательных чисел. При построении прямого, обратного, дополнительного кодов использовать битовые операции. Обеспечить эффективность кода по используемой памяти.
3. Для чисел типов double, float, long double вывести на экран их внутреннее представление в памяти компьютера (dump). Исходное значение вводится пользователем или задается программистом в тексте программы. Предусмотреть возможность обработки как положительных, так и отрицательных чисел. Для реализации вывода содержимого памяти использовать битовые операции. Обеспечить эффективность кода по используемой памяти.
4. Вывести в файл все простые числа из диапазона [0..1000000].Для отбора простых чисел использовать алгоритм "Решето Эратосфена" с битовым массивом. Числа из каждой сотни должны занимать в файле отдельную строку.

В процессе работы над лабораторной работой не допускается использование функции pow и алгоритмов перевода чисел в двоичную систему счисления.

Рекомендуемый вид консоли для 1, 2 и 3 частей:

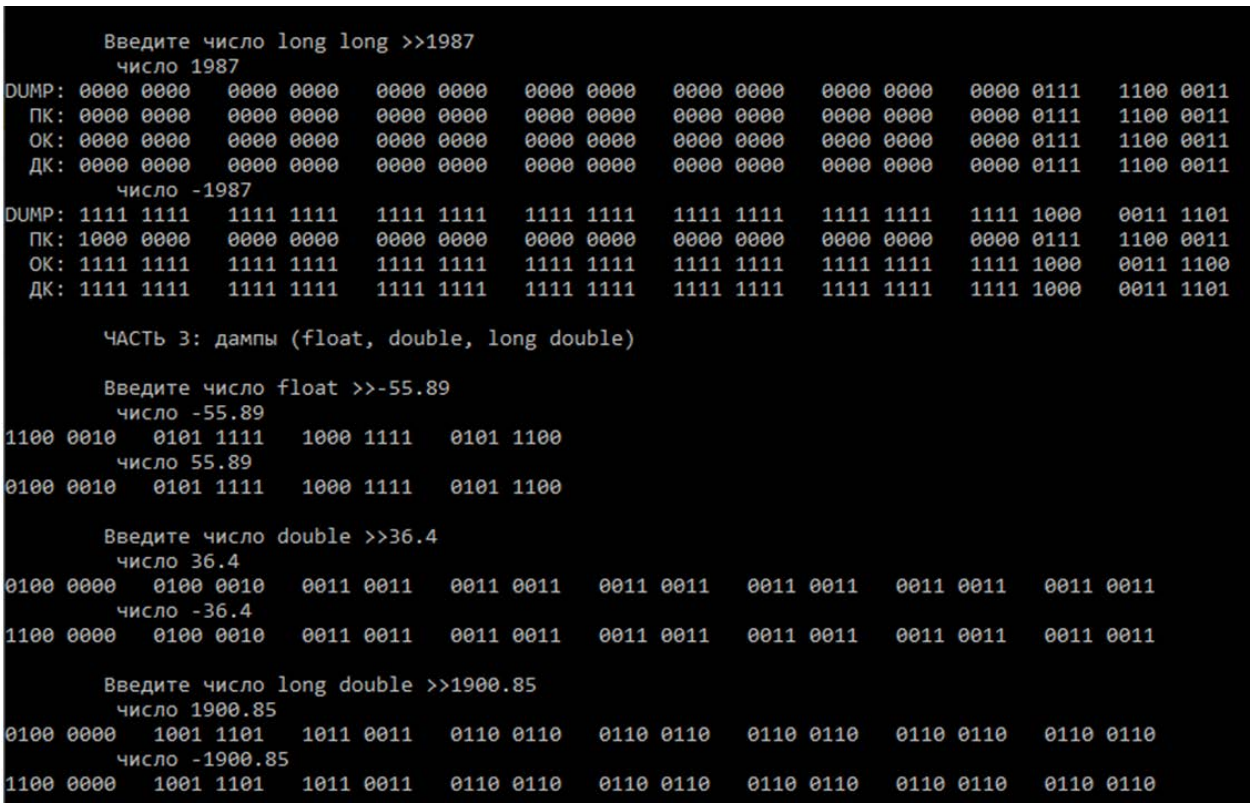
```
Введите число [-32000; 32000] >> 94
94: 0000 0000   0101 1110
1 - Установить бит в 1
2 - Сбросить бит в 0
3 - Опросить бит
4 - Изменить бит
0 - Выход
1
Введите номер бита >>5
2142: 0000 1000   0101 1110
1 - Установить бит в 1
2 - Сбросить бит в 0
3 - Опросить бит
4 - Изменить бит
0 - Выход
```

```
1
Введите номер бита >>1
-30626: 1000 1000   0101 1110
1 - Установить бит в 1
2 - Сбросить бит в 0
3 - Опросить бит
4 - Изменить бит
0 - Выход
2
Введите номер бита >>5
-32674: 1000 0000   0101 1110
1 - Установить бит в 1
2 - Сбросить бит в 0
3 - Опросить бит
4 - Изменить бит
0 - Выход
3
Введите номер бита >>9
-32674:         0
1 - Установить бит в 1
2 - Сбросить бит в 0
3 - Опросить бит
4 - Изменить бит
0 - Выход
4
Введите номер бита >>2
-16290: 1100 0000   0101 1110
1 - Установить бит в 1
2 - Сбросить бит в 0
3 - Опросить бит
4 - Изменить бит
0 - Выход
```

```
       ЧАСТЬ 2: дампы (short, int, long long)

       Введите число short >>-89
        число -89
DUMP: 1111 1111   1010 0111
  ПК: 1000 0000   0101 1001
  ОК: 1111 1111   1010 0110
  ДК: 1111 1111   1010 0111
        число 89
DUMP: 0000 0000   0101 1001
  ПК: 0000 0000   0101 1001
  ОК: 0000 0000   0101 1001
  ДК: 0000 0000   0101 1001

       Введите число int >>34
        число 34
DUMP: 0000 0000   0000 0000   0000 0000   0010 0010
  ПК: 0000 0000   0000 0000   0000 0000   0010 0010
  ОК: 0000 0000   0000 0000   0000 0000   0010 0010
  ДК: 0000 0000   0000 0000   0000 0000   0010 0010
        число -34
DUMP: 1111 1111   1111 1111   1111 1111   1101 1110
  ПК: 1000 0000   0000 0000   0000 0000   0010 0010
  ОК: 1111 1111   1111 1111   1111 1111   1101 1101
  ДК: 1111 1111   1111 1111   1111 1111   1101 1110
```



Лабораторная работа №4-5: Работа со строками

Цель: изучить основные функции и приемы работы со строками(как с массивом символов).

Задачи:

1. Реализовать в четырех вариантах каждую из шести функций работы со строками.
2. Снабдить программу системой тестирования и удобным интерфейсом.

Задание: Из функций strlen, strpos, strins, strdel, strcpy, strcmp, strcat* выбрать шесть, реализовать каждую в четырех вариантах. Варианты реализации между собой должны отличаться:

- по способу организации циклов;
- по подходу к обработке строк (указатели, массивы): хотя бы одна с указателем, хотя бы одна с массивом;
- алгоритмически.

Список входных параметров для одноименных функций должен совпадать.

Предусмотреть систему тестирования функций на различных примерах входных данных, в том числе на не корректных:

- пустые/непустые строки;
- отрицательные, нулевые, положительные индексы для функций strins, strdel, strcpy;
- выход за границы строки в любую из сторон;
- двойное и тройное вхождение (для функции strpos);

и т.д.

Рекомендуемый вид консоли:

strlen1("")=0
strlen2("")=0
strlen3("")=0
strlen4("")=0

strdel1("abcd",5,8)="abcd"
strdel2("abcd",5,8)="abcd"

strdel3("abcd",5,8)="abcd"
strdel4("abcd",5,8)="abcd"

На каждую функцию предусмотреть от двух до восьми тестовых примеров (в зависимости от выбранной функции). При выводе на экран все строки, в том числе пустые, заключаются в двойные кавычки. Ввод строк для тестовых примеров от пользователя необязателен.
При написании кода запрещено пользоваться функциями потокового ввода и вывода информации (cin, cout), использовать можно только стандартные функции ввода/вывода – printf, scanf.

Рекомендуемые сигнатуры и тесты для функций:

<table>
<tr><td>int Strlen(char* s)
– определение длины строки s (количество символов)</td><td colspan="3">2 теста:</td></tr>
<tr><td></td><td colspan="3">1. Пустая строка
2. Не пустая строка</td></tr>
<tr><td rowspan="2">int Strcmp(
char* s1,
char* s2)
– сравнение строк s1 и s2; возвращаемое значение: 0 – если s1=s2, 1 – если s1>s2, -1 – если s1<s2</td><td colspan="3">9 тестов:</td></tr>
<tr><td colspan="3">1. Сравнение с пустой строкой (только первая пуста, только вторая пуста, обе пусты)
2. Первая строка больше, но короче
3. Первая строка больше и длиннее
4. Первая строка больше и имеет такую же длину, как и вторая
5. Вторая строка больше и имеет такую же длину, как и первая
6. Вторая строка больше, но короче
7. Вторая строка больше и длиннее
8. Строки одинаковой длины, разница в последнем символе (первая больше, первая меньше)
9. Различие в первом символе</td></tr>
<tr><td rowspan="2">void Strdel(
char* src,
int p, int k)
– удаление из строки src символов: начиная с символа №p, k штук; src – исходная строка</td><td colspan="3">70 тестов (используются все комбинации):</td></tr>
<tr><td>1. src пуста
2. src не пуста</td><td>1. p<0
2. p=0
3. p>0, p<strlen(src)
4. p=strlen(src)
5. p>strlen(src)</td><td>1. p+k<0
2. p+k=0
3. p+k>0, p+k<strlen(src)
4. p+k=strlen(src)
5. p+k>strlen(src)
6. p=0
7. p=strlen(src)</td></tr>
<tr><td rowspan="2">void Strins(
char* src,
char* sub,
int p)
– вставка подстроки sub в строку src, начиная с символа №p; предполагается, что память под строку src</td><td colspan="3">20 тестов (используются все возможные комбинации):</td></tr>
<tr><td>1. src пуста
2. src не пуста</td><td>1. sub пуста
2. sub не пуста</td><td>1. p>0, p<strlen(src)
2. p<0 //в этом случае «приклеить» sub к src слева
3. p>strlen(src) //в этом случае «приклеить» sub к src справа
4. p=0
5. p=strlen(src)</td></tr>
</table>

выделена в достаточном количестве			
void Strcopy(char* dest, char* src, int p, int k) – копирование из строки src в строку dest p символов начиная с символа с номером k	***18 тестов (используются все комбинации):***		
	1. src пуста 2. src не пуста	1. p>=0, p<strlen(src) 2. p<0 3. p>=strlen(src)	1. k>0 2. k=0 3. k<0
int Strpos(char* src, char* sub) – позиция первого вхождения подстроки sub в строку src	***12 тестов (используются все комбинации):***		
	1. src пуста 2. src не пуста	1. sub пуста 2. sub не пуста, вхождений в src нет 3. sub не пуста, имеется единственное в хождение sub в src, начиная с символа k: А) k = 0 Б) 0<k<strlen(src)-strlen(sub) В) k= strlen(src)-strlen(sub) 4. sub не пуста, имеется несколько вхождений sub в src	
char* Strcut(int k, char* s1, char* s2, …) – объединение строк s1, s2, …, sk в одну строку; k – количество строк	***7 тестов (используются различные комбинации):***		
	1. k=0 2. k=1 3. k=2 4. k=3	1. среди строк s1,s2,s3…, sk есть пустые, есть не пустые 2. все строки s1, s2, s3, …, sk пусты 3. все строки s1, s2, s3, …, sk не пусты	

Возможны другие сигнатуры функций и количество тестов. При выборе количества параметров, их типов, а так же типа возвращаемого значения функций следует руководствоваться описанием стандартных функций языков C/C++/Pascal.

Лабораторная работа №6: Матрицы

Цель: изучить рациональные методы вычисления определителя и обратной матрицы, а также научиться решать системы линейных алгебраических уравнений (СЛАУ).

Задачи:

- изучение способов вычисления определителя и обратной матрицы;
- освоение вариантов обработки и сравнения вещественных чисел;
- освоение вариантов форматированного вывода матриц на экран.

Задание: Реализовать программу, позволяющую получать решение СЛАУ вида AX=B. Определитель матрицы и обратную матрицу рассчитывать методом Гаусса или Жордана-Гаусса, с помощью присоединенной матрицы.

Предусмотреть следующие возможности:

- задание размерности матриц пользователем;
- случайное разыгрывание и ручной ввод значений элементов матриц по выбору пользователя;

- вычисление определителя и обратной матрицы A, вывод полученных результатов на экран;
- наличие в матрице B нескольких столбцов;
- программа должна корректно обрабатывать входные данные для размерностей n<=15 (для этого не подойдут рекурсивные методы вычисления определителя и обратной матрицы);
- написать функцию умножения двух матриц произвольного размера с проверкой на определенность результата операции;
- написать функцию сложения двух матриц произвольного размера с проверкой на определенность результата операции;
- написать функцию вывода матрицы на экран;
- при выводе матриц использовать форматирование, выравнивание по столбцам;
- для ввода и вывода значений использовать функции printf/scanf;

Рекомендуемый вид консоли:

Введите размерность матрицы A: ->n
Задать значения элементов матрицы A вручную (1) или случайным образом (2)? ->1/2
Введите количество столбцов матрицы B: ->m
Задать значения элементов матрицы B вручную (1) или случайным образом (2)? ->1/2
A=... B=...
det(A)=...
A^-1=...
X=A^-1*B
X=...
Проверка:
A*X=...
A*X-B=...
A*A^-1=...
A^-1*A=...

При написании кода запрещено пользоваться функциями потокового ввода и вывода информации (cin, cout), использовать можно только стандартные функции ввода/вывода – printf, scanf.

Лабораторная работа №7: Алгоритм Флойда

Цель: Научиться работе с матрицей достижимости. Освоить алгоритм Флойда. Изучить основы алгоритмов, работающих с построением и анализом маршрутов.

Задачи:

- освоение алгоритма Флойда, изучение его особенностей;
- освоение визуальных возможностей консольных приложений при работе с выводом маршрутов и матриц.

Задание: По известной матрице достижимости определить матрицу кратчайших расстояний, а также вывести все составные маршруты. Предусмотреть следующие возможности:

- задание размерности матриц пользователем;
- случайное разыгрывание, чтение из файла и ручной ввод значений элементов матриц по выбору пользователя;
- вычисление матрицы кратчайших расстояний, вывод полученных результатов на экран;

- написать функцию вывода матрицы на экран;
- при выводе матриц использовать форматирование, выравнивание по столбцам;
- для ввода и вывода значений использовать функции printf/scanf;

Вид консоли должен быть следующим:

Введите размерность матрицы A: ->n
Задать значения элементов матрицы A вручную (1) или случайным образом (2)? ->1/2
Исходная матрица:

```
        1       2       3
      _____________
1     |   -     15      8
2     |   -1    -       11
3     |   4     -1      -
```

Матрица кратчайших расстояний:

```
        1       2       3
      _____________
1     |   -     15      8
2     |   15    -       11
3     |   4     19      -
```

Составные маршруты:

2→1(26):
2→3 (11) →1 (15)
3→2(23):
3→1 (4) →2 (19)

При написании кода лучше ограничиться от использования функций потокового ввода и вывода информации (cin, cout), рекомендуется использовать стандартные функции ввода/вывода – printf, scanf.

Рекомендуемый вид консоли:
Входная матрица:

0	100	0	0	0	3
100	0	400	4	0	0
0	400	0	500	7	5
0	4	500	0	2	0
0	0	7	2	0	68
3	0	5	0	68	0

Матрица кратчайших путей:

0	21	8	17	15	3
21	0	13	4	6	18
8	13	0	9	7	5
17	4	9	0	2	14
15	6	7	2	0	12
3	18	5	14	12	0

Матрица городов (при использовании соответствующего алгоритма):

-1	5	5	5	5	5
3	-1	3	3	3	3

5	4	-1	4	4	5
4	1	4	-1	4	4
2	3	2	3	-1	2
0	2	2	2	2	-1

Составные пути:
0→1(21):
0→ 5(3) → 2(5)→4(7)→3(2)→1(4)
0→2(8):
0→5(3)→2(5)
0→3(17):
0→5(3)→2(5)→4(7)→3(2)
0→4(15):
0→5(3)→2(5)→4(7)
1→0(21):
1→3(4)→4(2)→2(7)→5(5)→0(3)
1→2(13):
1→3(4)→4(2)→2(7)
1→4(6):
1→3(4)→4(2)
1→5(18):
1→3(4)→4(2)→2(7)→5(5)
2→0(8):
2→5(5)→0(3)
2→1(13):
2→4(7)→3(2)→1(4)
2→3(9):
2→4(7)→3(2)
3→0(17):
3→4(2)→2(7)→5(5)→0(3)
3→2(9):
3→4(2)→2(7)
3→5(14):
3→4(2)→2(7)→5(5)
4→0(15):
4→2(7)→5(5)→0(3)
4→1(6):
4→3(2)→1(4)
4→5(12):
4→2(7)→5(5)
5→1(18):
5→2(5)→4(7)→3(2)→1(4)
5→3(14):
5→2(5)→4(7)→3(2)
5→4(12):
5→2(5)→4(7)

Лабораторная работа №8: Длинная арифметика

Цель: Научиться работе с длинными числами, освоить основные подходы к обработке длинных чисел.

Задачи:
- разработка функций сложения двух длинных чисел, умножения двух длинных чисел, целочисленного деления длинного числа на длинное число, вычисления остатка от деления длинного числа на длинное число;
- разработка функций ввода/вывода, преобразования длинных чисел в формат для внутреннего хранения.

Задание: По двум заданным пользователем с клавиатуры длинным числам определить результат арифметических операций с ними.
Хранение длинных чисел предусмотреть в виде десятичного дополнительного кода. При организации деления осуществлять все необходимые проверки определенности результата. При разработке программы отдельными функциями должны быть оформлены:
- считывание и формирование в памяти длинного числа;
- вывод на экран длинного числа;
- смена знака длинного числа;
- умножение двух длинных чисел;
- сложение двух длинных чисел;
- целочисленное деление длинных чисел: реализовать имитацией деления столбиком.

Разработать систему тестирования, включить следующие примеры:
- два числа длины >=15 знаков каждое;
- одно число длины >=15, другое – двухзначное;
- два четырехзначных числа.

Рекомендуемый видконсоли:
Введите a: 135624514562345
Введитеb: 2213452
a+b = 135624516775797
a – b =135624512348893
b – a = –135624512348893
a*b =300198353007042664940
a div b = 61272850
b div a = 0
a mod b = 2184145
b mod a =2213452

Лабораторная работа №9: Обратная польская запись

Цель: освоить методы работы со стеком.
Задачи:

- ознакомиться с алгоритмом формирования обратной польской записи (ОПЗ) по входному выражению, содержащему числа, знаки арифметических операций (‘+’, ‘-’, ‘*’, ‘/’) и скобки (‘(‘, ‘)’); реализовать программно алгоритм формирования ОПЗ;
- ознакомиться и реализовать программно алгоритм обработки выражения в виде ОПЗ с использованием стека.

Задание: На вход задачи поступает выражение, содержащее знаки арифметических операций, числа и закрывающие и открывающие скобки (при работе с многозначными числами допускается наличие пробелов во входной последовательности).
Требуется по имеющейся входной последовательности сформировать ОПЗ (1 этап), а затем по полученной ОПЗ рассчитать результат (2 этап).

На обоих этапах решения задачи использовать стек: на первом этапе – для хранения чисел из исходной строки, на втором этапе – для хранения чисел из ОПЗ и результата. Полученную на 1 этапе ОПЗ вывести на экран.
В каждом случае обнаружения некорректности входной строки генерировать сообщение пользователю.
Допустимо решать задачу в предположении, что исходная последовательность содержит лишь однозначные числа и не содержит унарного знака «минус».

Дополнительно:
- вывести промежуточные вычисления (см. в примере рекомендуемого вида консоли, метод – по предпочтению) (+5 баллов);
- реализовать работу унарного знака «минус» (+7 баллов)
- реализовать работу всей программы для многозначных чисел (+10 баллов)

Рекомендуемый вид консоли:
Введите выражение:
(5-3)*6*2+9*5-3*(1-8)
ОПЗ:
5 3 – 6 * 2 * 9 5 * + 3 1 8 - * -

Дополнительно:
*= 2 6 * 2 * 45 + 3 (-7) * -*
*= 12 2 * 45 + (-21) –*
= 24 45 + (-21) –
= 69 (-21) –
= 90

Результат работы:
90

Лабораторная работа №10: Задача о библиотекаре

Цель: освоение некоторых специфических алгоритмов обработки массивов и методов работы с текстовыми файлами.
Задачи:
- ознакомиться с методами чтения и записи файлов, содержащих текстовую информацию;
- разобрать очередной подход к обработке массивов.

Задание: Библиотекарь – пожилой мужчина, работающий в библиотеке. На любимой книжной полке библиотекаря расположены N<201 книг. Ежедневно (библиотека работает даже в выходные дни!) нерадивые читатели устремляются к любимой книжной полке библиотекаря и хватают оттуда книги, после чего в течение определенного времени читают их, и к концу дня ставят обратно на полку в случайном порядке.

Библиотекарь не может допустить, чтобы книги в конце дня располагались на полке не в алфавитном порядке, поэтому он считает своим долгом переставить их для упорядочивания. В молодости библиотекарь мог без труда и с большим удовольствием упорядочить книги хоть на целом стеллаже! Но теперь… Руки-то уж не те. Библиотекарь желает минимизировать количество своих действий, потраченных для упорядочивания книг на любимой полке. Одно его действие заключается в том, что библиотекарь

извлекает книгу с полки и ставит ее на полку в другое место, между какими-либо двумя другими книгами.

Книги на полке могут быть и одинаковые.

Исходные данные представлены в виде текстовой информации о книгах, которая находится в файле input.txt.

Рекомендуемый вид консоли:

Исходный файл:

Пушкин А.С. «Евгений Онегин»
Тургенев И.С. «Отцы и дети»
Логинов С.В. «Свет в окошке»
Пушкин А.С. «Евгений Онегин»
Стругацкий Б.Н., Стругацкий А.Н. «Понедельник начинается в субботу»
Янссон Т.С. «Волшебная зима»
Тургенев И.С. «Отцы и дети»
Толстой Л.Н. «Война и мир»

Отсортированная информация:

2. Пушкин А.С. «Евгений Онегин»
5. Тургенев И.С. «Отцы и дети»
1. Логинов С.В. «Свет в окошке»
2. Пушкин А.С. «Евгений Онегин»
3. Стругацкий Б.Н., Стругацкий А.Н. «Понедельник начинается в субботу»
6. Янссон Т.С. «Волшебная зима»
5. Тургенев И.С. «Отцы и дети»
4. Толстой Л.Н. «Война и мир»

1. Логинов С.В. «Свет в окошке»
2. Пушкин А.С. «Евгений Онегин»
2. Пушкин А.С. «Евгений Онегин»
3. Стругацкий Б.Н., Стругацкий А.Н. «Понедельник начинается в субботу»
4. Толстой Л.Н. «Война и мир»
5. Тургенев И.С. «Отцы и дети»
5. Тургенев И.С. «Отцы и дети»
6. Янссон Т.С. «Волшебная зима»

Массив:

2	5	1	2	3	6	5	4
1	2	1	2	3	4	4	4

Библиотекарю необходимо выполнить 4 действия.
Книги, которые останутся на месте:

2. Пушкин А.С. «Евгений Онегин»
5. Тургенев И.С. «Отцы и дети»
1. Логинов С.В. «Свет в окошке»
2. Пушкин А.С. «Евгений Онегин»
3. Стругацкий Б.Н., Стругацкий А.Н. «Понедельник начинается в субботу»
6. Янссон Т.С. «Волшебная зима»
5. Тургенев И.С. «Отцы и дети»
4. Толстой Л.Н. «Война и мир»

Лабораторная работа №11: методы решения оптимизационных задач выбора. Задача о загрузке рюкзака.

Цель: знакомство с принципами выделения кода в функцию. Освоение подходов к решению оптимизационных задач: полный перебор, методы с отсечением.

Задачи:

- освоить метод полного перебора двоичных цепочек с последующей обработкой;
- освоить метод ветвей и границ и другие методы с отсечением.

Задание: Имеется рюкзак фиксированной емкости R и набор грузов $a=\{a_i\}$, i=1..n, которые необходимо разместить в рюкзаке. Данные таковы, что все грузы в рюкзак одновременно не поместятся, поэтому нужно выбрать наиболее выгодные варианты заполнения рюкзака. В версии задач V3 и V4 кроме указанных данных так же заданы стоимости грузов $c=\{c_i\}$, i=1..n.

Наиболее выгодными вариантами заполнения рюкзака считаются те, которые обеспечивают наименьшее свободное пространство (V1, V2), либо те, которые обеспечивают максимальную стоимость выборки (V3, V4).

Требуется найти количество лучших решений и вывести на экран не более 100 из них. Задачу необходимо решить в четырех различных вариантах. Все допустимые выборки должны удовлетворять следующим условиям:

1) суммарный вес положенных в рюкзак объектов не превышал заданного порога R (вместимости рюкзака);

2) незаполненная часть рюкзака R- $\sum_{i=1}^{N} вес\_i-го\_объекта$ достигала минимума.

V1. Заданы емкость рюкзака R и веса грузов $a=\{a_i\}$, i=1..n. Осуществить полный перебор двоичных векторов $v=\{v_i\}$, $v_i \in \{0; 1\}$, i=1..n.

V2. Заданы емкость рюкзака R и веса грузов $a=\{a_i\}$, i=1..n. Осуществить перебор допустимых выборок с отсечением.

V3. Заданы емкость рюкзака R, веса грузов $a=\{a_i\}$, и цены $c=\{c_i\}$, i=1..n. Осуществить полный перебор двоичных векторов $v=\{v_i\}$, $v_i \in \{0; 1\}$, i=1..n.

V4. Заданы емкость рюкзака R и веса грузов $a=\{a_i\}$, и цены $c=\{c_i\}$, i=1..n. Осуществить перебор допустимых выборок с отсечением, используя сортировку исходных данных по убыванию удельной стоимости грузов и данные о максимальной стоимости остатка рюкзака. Использовать алгоритм с возвратами (обход дерева в глубину).

Лабораторная работа №12: Бинарный поиск.