

# Компиляторы

—

МУ ПО ЛАБОРАТОРНЫМ РАБОТАМ

# Лабораторная работа № 3.1

Напишите регулярные определения для следующих языков.

- а) Все строки из строчных букв, содержащие пять гласных, а, е, і, о, и, в указанном порядке.
- б) Все строки из строчных букв, в которых буквы находятся в возрастающем лексикографическом порядке.
- в) Комментарии, представляющие собой строки, заключенные в /\* и \*/ , без промежуточных символов \*/ (кроме случаев, когда они заключены в двойные кавычки).
- г) Все строки из неповторяющихся цифр. Указание: попробуйте сначала решить задачу для нескольких цифр, например для {0, 1, 2}.
- д) Все строки из цифр, причем в строке может повторяться не более одной цифры.
- е) Все строки из а и b, в которых четное количество а и нечетное — b.

## Лабораторная работа № 3.2

В SQL ключевые слова и идентификаторы нечувствительны к регистру. Напишите программу на языке Lex, которая распознает ключевые слова SELECT, FROM и WHERE (с любыми сочетаниями верхних и нижних регистров) и токен ID, который в данном упражнении может быть любой последовательностью букв и цифр, начинающейся с буквы. Вносить идентификаторы в таблицу символов не требуется, но следует указать, чем именно функция для внесения в таблицу символов отличается от таковой для идентификаторов, чувствительных к регистру, как на следующем слайде.

```

%{
/* Определения именованных констант
LT, LE, EQ, NE, GT, GE,
IF, THEN, ELSE, ID, NUMBER, RELOP */
%}

/* Регулярные определения */
delim    [ \t\n]
ws       {delim}+
letter   [A-Za-z]
digit    [0-9]
id       {letter}({letter}|{digit})*
number   {digit}+(\.{digit}+)?(E[+-]?{digit}+)?

%%

{ws}     {/* Нет действий и выхода из функции */}
if       {return(IF);}
then     {return(THEN);}
else     {return(ELSE);}
{id}     {yyval = (int) installID(); return(ID);}
{number} {yyval = (int) installNum(); return(NUMBER);}
"<"     {yyval = LT; return(RELOP);}
"<="    {yyval = LE; return(RELOP);}
"="      {yyval = EQ; return(RELOP);}
">"     {yyval = NE; return(RELOP);}
">="    {yyval = GE; return(RELOP);}

%%

int installID() {/* Функция для внесения лексемы
(на первый символ которой указывает
указатель yyltext и длина которой равна
yyleng) в таблицу символов. Возвращает
указатель на соответствующую запись
таблицы символов */
}

int installNum() {/* Аналогична функции installID, но
помещает числовые константы в
отдельную таблицу */
}

```

# Лабораторная работа № 6.1

Массив действительных чисел  $A [i, j, k]$  имеет индекс  $i$  со значениями в диапазоне от 1 до 4, индекс  $j$  со значениями от 0 до 4 и индекс  $k$  со значениями от 5 до 10. Размер каждого действительного числа — 8 байт. Предположим, что массив  $A$  хранится построчно, начиная с байта 0. Найдите положения следующих элементов массива:

# Лабораторная работа № 6.2

Транслируйте данные выражения в

а) синтаксическое дерево;

б) четверки;

в) тройки;

г) косвенные тройки.

i)  $a = b[i] + c[j]$

ii)  $a[i] = b * c - b * d$

iii)  $x = f(y+1) + 2$

iv)  $x = *p + \&y$

# Лабораторная работа № 9.1

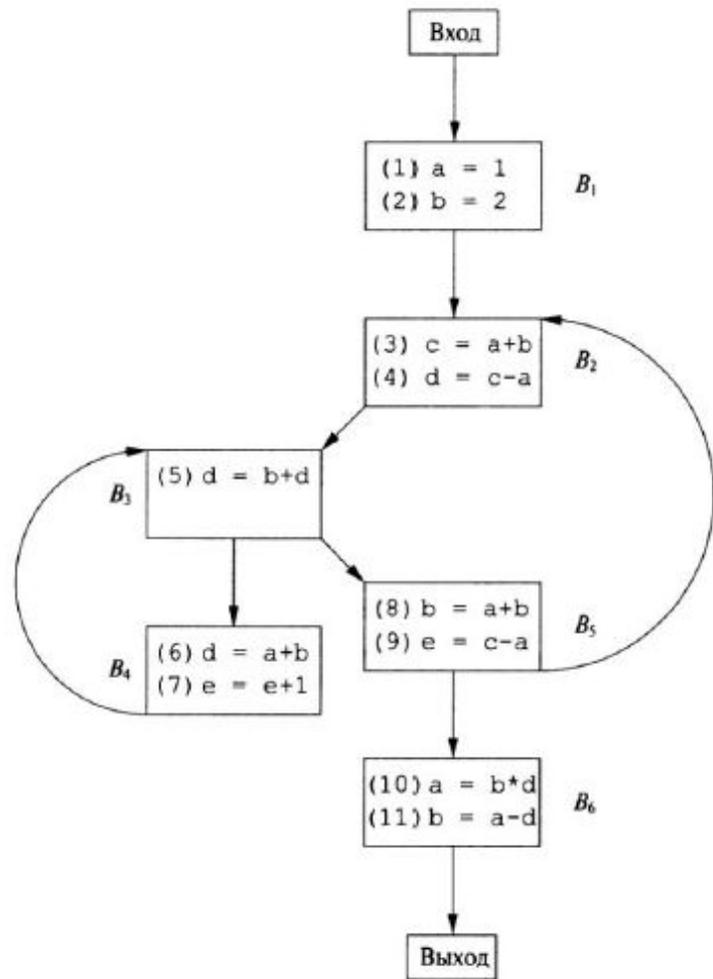
На следующем слайде приведен промежуточный код для вычисления скалярного произведения векторов  $A$  и  $B$ . Оптимизируйте насколько возможно этот код путем устранения общих подвыражений, снижения стоимости и устранения переменных индукции.

```
    dp = 0.  
    i = 0  
L:   t1 = i*8  
     t2 = A[t1]  
     t3 = i*8  
     t4 = B[t3]  
     t5 = t2*t4  
     dp = dp+t5  
     i = i+1  
     if i<n goto L
```

# Лабораторная работа № 9.2

Для графа на следующем слайде выполните следующее.

- а) Вычислите отношение доминирования.
- б) Постройте дерево доминаторов
- в) Найдите для каждого узла его непосредственный доминатор.
- г) Найдите упорядочение вглубь графа потока.
- д) Укажите в вашем ответе к п. г наступающие, отступающие, поперечные ребра и ребра дерева.
- е) Является ли данный граф потока приводимым?
- ж) Вычислите глубину графа потока.
- з) Найдите естественные циклы в графе потока.



# Лабораторная работа № 12.1

На следующем слайде приведена программа на языке программирования C с двумя указателями на функции —  $p$  и  $q$ .  $N$  — константа, которая может быть меньше или больше 10. Заметим, что в результате в программе получится бесконечная последовательность вызовов, но в данном случае нас это волновать не должно.

- а) Укажите все точки вызова программы.
- б) Укажите для каждой точки вызова, на что указывает  $p$ ? На что указывает  $q$ ?
- в) Изобразите граф вызовов для данной программы.
- г) Опишите все строки вызовов для функций  $f$  и  $g$ .

```
int (*p)(int);
int (*q)(int);

int f(int i) {
    if (i < 10)
        {p = &q; return (*q)(i);}
    else
        {p = &f; return (*p)(i);}
}

int g(int j) {
    if (j < 10)
        {q = &f; return (*p)(j);}
    else
        {q = &g; return (*q)(j);}
}

void main() {
    p = &f;
    q = &g;
    (*p)((*q)(N));
}
```

# Лабораторная работа № 12.2

Правило

$$p(X, Y) :- q(X, Z) \ \& \ r(Z, W) \ \& \ \text{NOT } p(W, Y)$$

является частью большей Datalog-программы  $P$  .

- а) Укажите заголовок, тело и подцели данного правила.
- б) Какие предикаты определенно являются IDB-предикатами программы  $P$ ?
- в) Какие предикаты определенно являются EDB-предикатами программы  $P$ ?
- г) Является ли это правило безопасным?
- д) Стратифицирована ли программа  $P$ ?