**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИИ**  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»**  
**(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №6

Дисциплина «Архит. ВС»

Тема «Логические команды и команды сдвига»

Выполнил: ст. группы ПВ-31  
Ковалев Павел Александрович

Проверил: Осипов Олег Васильевич

**Белгород 2021**

**Вариант 9**

**Цель работы:** изучение команд поразрядной обработки данных.

**Задания для выполнения к работе**

1. Написать программу для вывода чисел на экран согласно варианту задания. При выполнении задания №1 все числа считать беззнаковыми. Написать и использовать функцию output(a) для вывода числа a на экран или в файл. Функция должна удовлетворять соглашению о вызовах. В функцию для вывода output передавать в качестве аргумента переменную размерности 32 или 64 бита, которой достаточно для хранения числа. К примеру, если в задании число указано как 15-разрядное, то аргументом функции должно быть число размером двойное слово, если 40-разрядное, то учетверённое слово. Функция должна выводить столько разрядов числа, сколько указано в задании, даже если старшие разряды равны нулю. Не допускается прямой перебор всех чисел с проверкой, удовлетворяет ли оно условию вывода (за исключением вариантов № 8, 12, 13). Числа выводить в порядке, который является удобным. Проверить количество выведенных чисел с помощью формул комбинаторики. В отчёт включить вывод формул и результаты работы программы.
2. Написать подпрограмму для умножения (multiplication) или деления (division) большого целого числа на 2n (в зависимости от варианта задания) с использованием команд сдвига. Подпрограммы должны иметь следующие заголовки:

multiplication(char\* a, int n, char\* res);

division(char\* a, int n, char\* res).

Входные параметры: a – адрес первого числа в памяти, n – степень двойки.

Выходные параметры: res – адрес массива, куда записывается результат. В случае операции умножения, для массива res зарезервировать в два раза больше памяти, чем для множителей a и b. Числа a, b, res вывести на экран в 16-ричном виде. Подобрать набор тестовых данных для проверки правильности работы подпрограммы.

**Выполнение работы**

**Вывод задания**

1. AAAAAAAAAAA3
2. AAAAAAAAAA3A
3. AAAAAAAAA3AA
4. AAAAAAAA3AAA
5. AAAAAAA3AAAA
6. AAAAAA3AAAAA
7. AAAAA3AAAAAA
8. AAAA3AAAAAAA
9. AAA3AAAAAAAA
10. AA3AAAAAAAAA
11. A3AAAAAAAAAA
12. 3AAAAAAAAAAA
13. AAAAAAAAAAA5
14. AAAAAAAAAA5A
15. AAAAAAAAA5AA
16. AAAAAAAA5AAA
17. AAAAAAA5AAAA
18. AAAAAA5AAAAA
19. AAAAA5AAAAAA
20. AAAA5AAAAAAA
21. AAA5AAAAAAAA
22. AA5AAAAAAAAA
23. A5AAAAAAAAAA
24. 5AAAAAAAAAAA
25. AAAAAAAAAAA7
26. AAAAAAAAAA7A
27. AAAAAAAAA7AA
28. AAAAAAAA7AAA
29. AAAAAAA7AAAA
30. AAAAAA7AAAAA
31. AAAAA7AAAAAA
32. AAAA7AAAAAAA
33. AAA7AAAAAAAA
34. AA7AAAAAAAAA
35. A7AAAAAAAAAA
36. 7AAAAAAAAAAA
37. AAAAAAAAAAAF
38. AAAAAAAAAAFA
39. AAAAAAAAAFAA
40. AAAAAAAAFAAA
41. AAAAAAAFAAAA
42. AAAAAAFAAAAA
43. AAAAAFAAAAAA
44. AAAAFAAAAAAA
45. AAAFAAAAAAAA
46. AAFAAAAAAAAA
47. AFAAAAAAAAAA
48. FAAAAAAAAAAA

Задания 2

Входное значение Вывод функции 1 400000000002 800000000004

2 111111111111 222222222222

3 12345678 24691356

Task1.asm

.386

.model flat, stdcall

option casemap: none

include c:\masm32\include\windows.inc

include c:\masm32\include\kernel32.inc

include c:\masm32\include\user32.inc

include c:\masm32\include\msvcrt.inc

includelib c:\masm32\lib\user32.lib

includelib c:\masm32\lib\kernel32.lib

includelib c:\masm32\lib\msvcrt.lib

.DATA

alphabet db 3h,5h,7h,0Fh

count\_format db "%d. ", 0

half\_format db "%X", 0

number\_format db "%X",0Ah

new\_line\_format db 13, 10, 0

empty\_num dd 0AAAAAAAAh

half\_num dw 0AAAAh

.CODE

safe\_print proc

push eax

push ebx

push ecx

push edx

push ebp

mov eax,[esp+24]

mov ebx,[esp+28]

;форматная строка

push ebx

;операнд

push eax

call crt\_printf

pop eax

pop eax

pop ebp

pop edx

pop ecx

pop ebx

pop eax

ret 8

safe\_print endp

START:

xor eax,eax

xor ecx,ecx

xor ebp,ebp

mov ebp,1

mov ecx,1

loop\_start:

;вложенный цикл

mov bl,alphabet[ecx-1]

push ecx

mov ecx,7

;установка первого символа

xor edx,edx

mov dx,half\_num

mov eax,empty\_num

shl eax, 4

add al,bl

;вывод первого числа

push ebp

push offset count\_format

call safe\_print

inc ebp

push edx

push offset half\_format

call safe\_print

push eax

push offset number\_format

call safe\_print

ex\_loop:

shl eax, 4

add al,000Ah

push ebp

push offset count\_format

call safe\_print

inc ebp

push edx

push offset half\_format

call safe\_print

push eax

push offset number\_format

call safe\_print

loop ex\_loop

mov ecx,3

mov eax,empty\_num

shl edx,4

add dl,bl

and edx,0000FFFFh

push ebp

push offset count\_format

call safe\_print

inc ebp

push edx

push offset half\_format

call safe\_print

push eax

push offset number\_format

call safe\_print

ex\_loop2:

shl edx,4

add dl,000Ah

and edx,0000FFFFh

push ebp

push offset count\_format

call safe\_print

inc ebp

push edx

push offset half\_format

call safe\_print

push eax

push offset number\_format

call safe\_print

loop ex\_loop2

pop ecx

inc ecx

cmp ecx,5

jl loop\_start

END START

Task2.asm

.386

.model flat, stdcall

option casemap: none

include c:\masm32\include\windows.inc

.DATA

format db "%x%x"

big\_number dd 2d,0d,0d,0d,0d,0d,0d,0d,0d,0d,4d

.CODE

mult\_arr proc

;x=x<<n

;4 байта - n

;4 байта - x

;4 байта - адрес возврата

push eax

push ebx

push ecx

push edx

;16 байт - сохранение переменных

xor edx,edx

mov ecx,[esp+20]

outer\_loop:

push ecx

mov ecx,0

clc

inner\_loop:

;адрес начала массива x

mov ebx,[esp+28]

;x[ecx]

mov eax,dword ptr [ecx\*4+ebx]

;прибавить CF

mov edx,0

adc edx,0

or eax,edx

;сдвиг влево (старший бит в CF)

shl eax,1

;запись обратно

mov [ecx\*4+ebx],eax

inc ecx

cmp ecx,11

jl inner\_loop

pop ecx

loop outer\_loop

pop edx

pop ecx

pop ebx

pop eax

ret 8

mult\_arr endp

START:

push offset big\_number

push 1

call mult\_arr

END STAR