

Міністерство освіти і науки, молоді та спорту України

Національний аерокосмічний університет
ім. М.Є. Жуковського

Кафедра 503

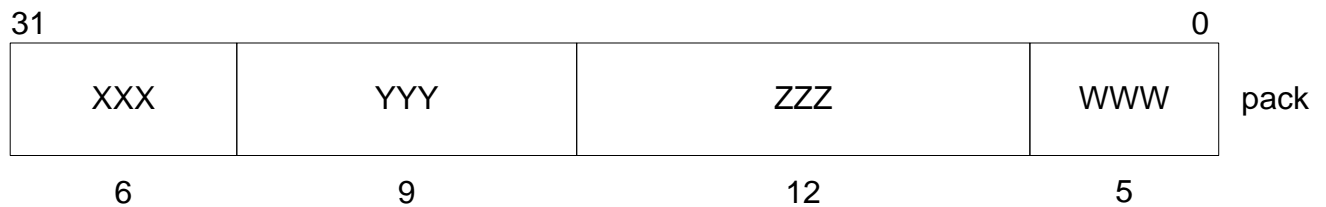
Лабораторна робота № 4
з дисципліни
«Архітектура комп'ютерів»
Тема: «Распаковка бітових груп»

Варіант № 37

Виконав:	ст. гр. 525д Петренко Р.Е.
Перевірив:	ст. викладач Дужий В.І.

Харків 2012

Задание. Дано 32-битное целое беззнаковое число *pack*, которое содержит четыре битовые группы, хранящие независимую информацию.

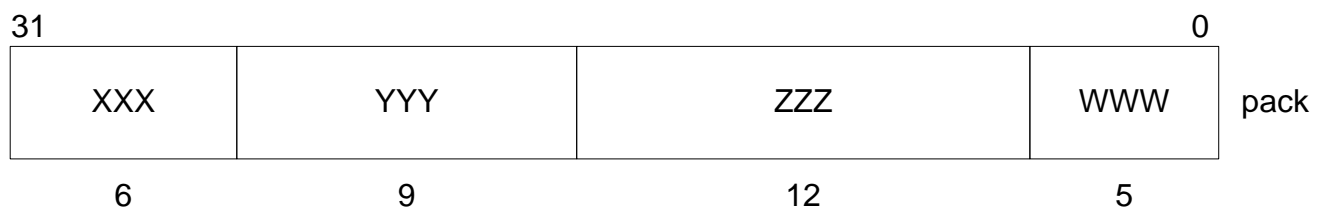


Написать программу программы на С и на ассемблере, которая *распаковывает* двоичные группы, содержащиеся в переменной *pack*. Биты каждого упакованного поля должны располагаться в младших разрядах соответствующей переменной, старшие разряды которой должны содержать нулевые биты. Для размещения каждого поля использовать стандартную битовую группу минимальной длины (байт, слово или длинное слово).

1. Исходные данные.

pack – переменная, длинное целое беззнаковое число.

Формат исходных данных.

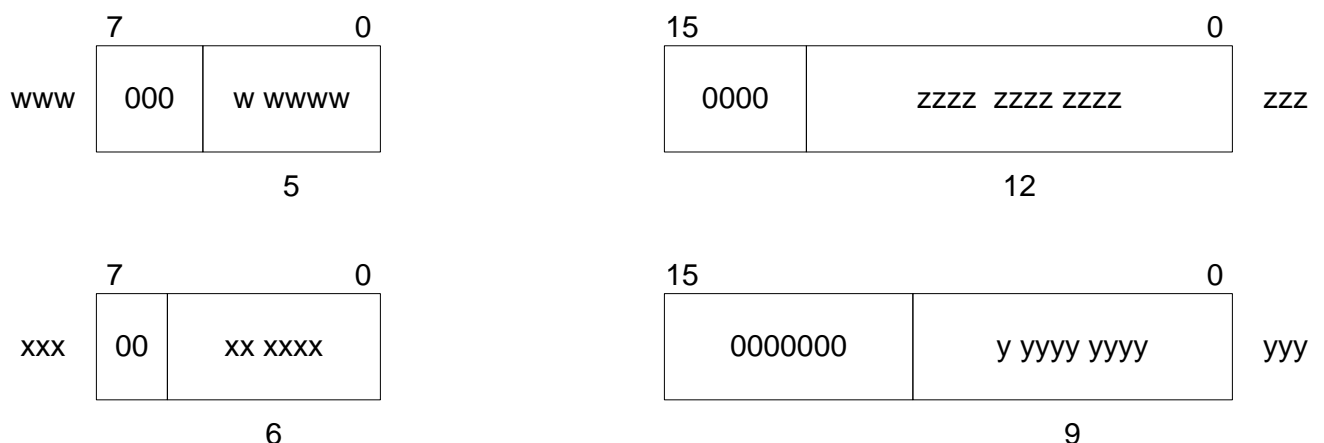


2. Требуемый результат.

x, *w* – переменные, типа *unsigned char*, длиной байт;

y, *z* - переменные, типа *unsigned short*, длиной слово.

Формат требуемого результата.



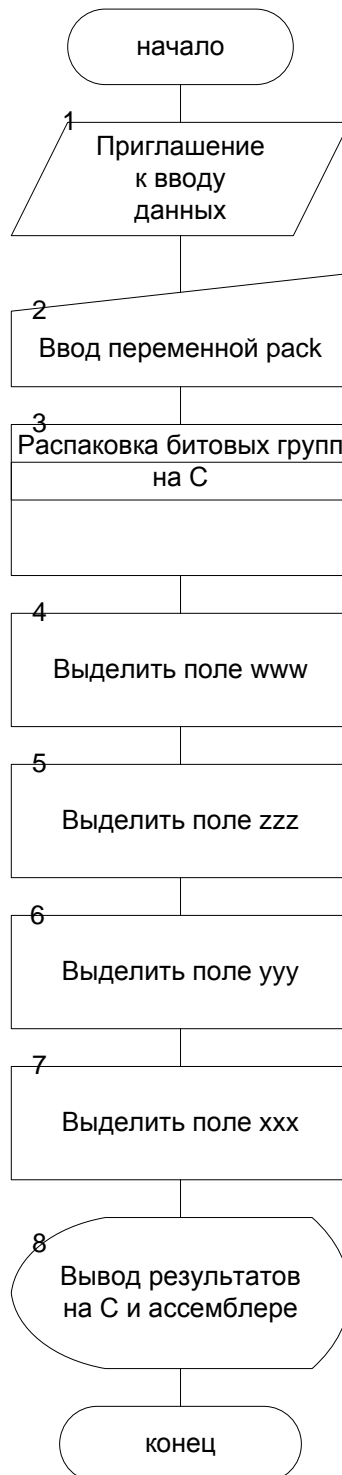
3. Описание алгоритма на псевдокоде.

Ввести исходное число с упакованными полями;

Выделить из числа поле *www* на С;

Выделить из числа поле *zzz* на С;

Выделить из числа поле ууу на С;
Выделить из числа поле ххх на С;
Выделить из числа поле www на ассемблере;
Выделить из числа поле zzz на ассемблере;
Выделить из числа поле ууу на ассемблере;
Выделить из числа поле ххх на ассемблере;
Вывести значения переменных на С;
Вывести значения переменных на ассемблере.



4. Текст программы с комментариями.

```
//+=====
// File unpack.cc
// Распаковка битовых групп
//
// Эта программа распаковывает битовые группы из целого беззнакового числа
//
// (С) Дужий В.И., 2012
//
// Входные данные:
// Длинное целое число, которое содержит несколько битовых групп
//
//          31-----+-----+-----+-----0
//          ! xxx ! yyy ! zzz ! www !  value
//          +-----+-----+-----+-----+
//          6         9         12        5
//
// Выходные данные:
//          7-----5-----0
//          ! 00 !  xxx !   xxx
//          +----+-----+
//          6
//          15---8-----0
//          ! 00 !  yyy !   yyy
//          +----+-----+
//          9
//          15---11-----0
//          ! 00 !  zzz !   zzz
//          +----+-----+
//          12
//          7-----4-----0
//          ! 00 !  www !   www
//          +----+-----+
//          5
//-----
#include <iostream>
#include <iomanip>
using namespace std;

unsigned long  value;
unsigned char  xxx, www, xxx_a, www_a;
unsigned short yyy, zzz, yyy_a, zzz_a;

int main()
{
    printf("\n\t\t(С) Дужий В.И., 2012");
    printf("\n\t\tРаспаковка битовых групп");
    for (;;)
    {
        printf("\n\tРаспаковать 32-битовое число Value");
        printf("\nПожалуйста, введите 8 16-ых цифр (например, 5a9db8e4) : ");
        scanf("%x",&value);
//===== С =====
// Выделить битовые группы из числа Value
        www = value & 0x1f;          // выделить 5 младших битов(4-0) из Value
        zzz = (value >> 5) & 0xfff; // выделить 12 битов (16-5) из Value
        yyy = (value >> 17) & 0x1ff; // выделить 9 битов (25-17) из Value
        xxx = (value >> 26) & 0x3f ; // выделить 6 битов (31-26) из Value
```

```
//===== Assembler =====
// Выделить битовые группы из числа Value
__asm{
    mov     eax,value
// выделить 5 младших битов(4-0) из Value
    mov     www_a,al
    and     www_a,0x1f
    shr     eax,5
// выделить 12 битов (16-5) из Value
    mov     zzz_a,ax
    and     zzz_a,0xffff
    shr     eax,12
// выделить 9 битов (25-17) из Value
    mov     yyy_a,ax
    and     yyy_a,0x1fff
    shr     eax,$9
// выделить 6 битов (31-26) из Value
    mov     xxx_a,al
    and     xxx_a,0x3f
};
// Форматный вывод результатов
cout << hex
    << "Битовая группа XXX (C++): " << (int) xxx
    << "\nБитовая группа YYY (C++): " << (int) yyy
    << "\nБитовая группа ZZZ (C++): " << (int) zzz
    << "\nБитовая группа WWW (C++): " << (int) www
    << "\n    Битовая группа XXX (Asm): " << (int) xxx_a
    << "\n    Битовая группа YYY (Asm): " << (int) yyy_a
    << "\n    Битовая группа ZZZ (Asm): " << (int) zzz_a
    << "\n    Битовая группа WWW (Asm): " << (int) www_a
    << endl;
};
return 0;
}
```

5. Тестовые примеры.

Номер	Исходное число, value	Ожидаемый результат				Полученный результат				Цель теста
		xxx	yyy	zzz	www	xxx	yyy	zzz	www	
1	FFFF FFFF	3f	1ff	fff	1f					Все биты равны 1
2	AAAA AAAA	2a	155	555	a					Чередующиеся 1 и 0
3	5555 5555	15	aa	aaa	15					Чередующиеся 0 и 1
4	FC000000	3f	0	0	0					Все 1 в битовой группе xxx
5	1F	0	0	0	1f					Все 1 в битовой группе www
6	3FE0000	0	1ff	0	0					Все 1 в битовой группе ууу
7	1FFE0	0	0	fff	0					Все 1 в битовой группе zzz
8	1234 5678									Произвольные биты
9	1A2B 3C4D									Произвольные биты