

Міністерство освіти і науки, молоді та спорту України

Національний аерокосмічний університет  
ім. М.Є. Жуковського

Кафедра 503

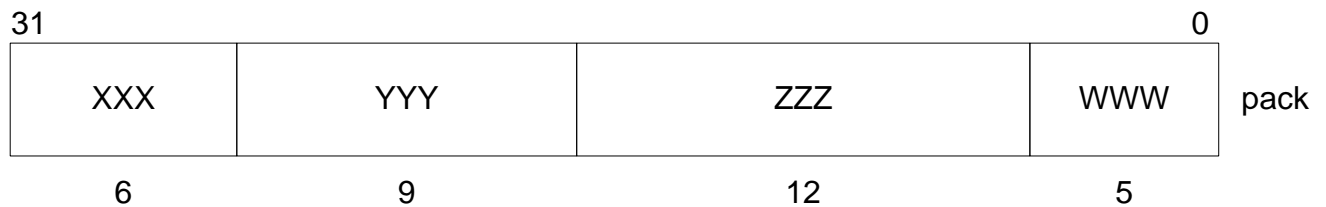
Лабораторна робота № 5  
з дисципліни  
«Архітектура комп'ютерів»  
Тема: «Упаковка бітових груп»

Варіант № 37

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

Харків 2012

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



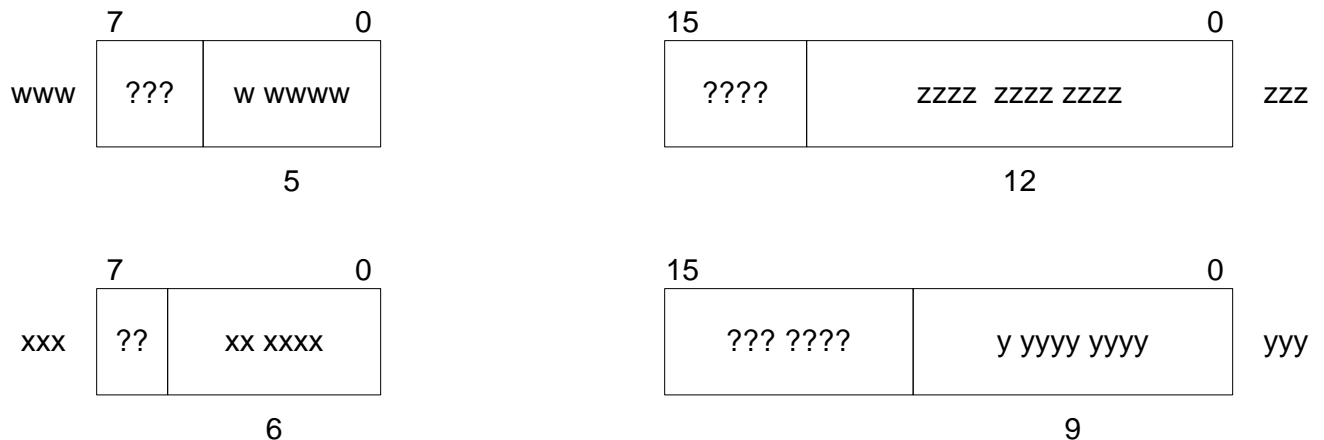
Написать программу программы на C и на ассемблере, которая *упаковывает* группы битов, содержащиеся в изолированных переменных `x`, `y`, `z` и `w`, в одно 32-битовое целое беззнаковое число `pack`.

## 2. Исходные данные.

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

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

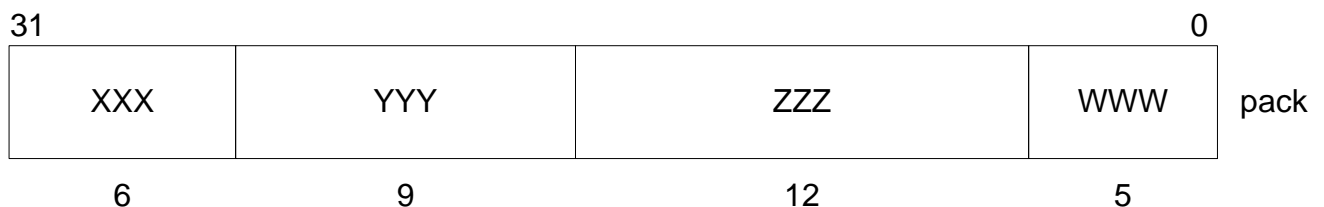
### Формат исходных данных.



## 3. Требуемый результат.

`pack` – переменная, длинное целое беззнаковое число.

### Формат требуемого результата.



## 4. Описание алгоритма на псевдокоде.

Ввести исходные переменные `xxx`, `yuy`, `zzz`, `www`;

Очистить старшие биты в каждой исходной переменной на C;

Объединить поле `xxx` и число `pack` на C;

Объединить поле `yuy` и число `pack` на C;

Объединить поле `zzz` и число `pack` на C;

Объединить поле `www` и число `pack` на `C`;

Очистить старшие биты в каждой исходной переменной на ассемблере;

Очистить результат на ассемблере;

Объединить поле `xxx` и число `pack` на ассемблере;

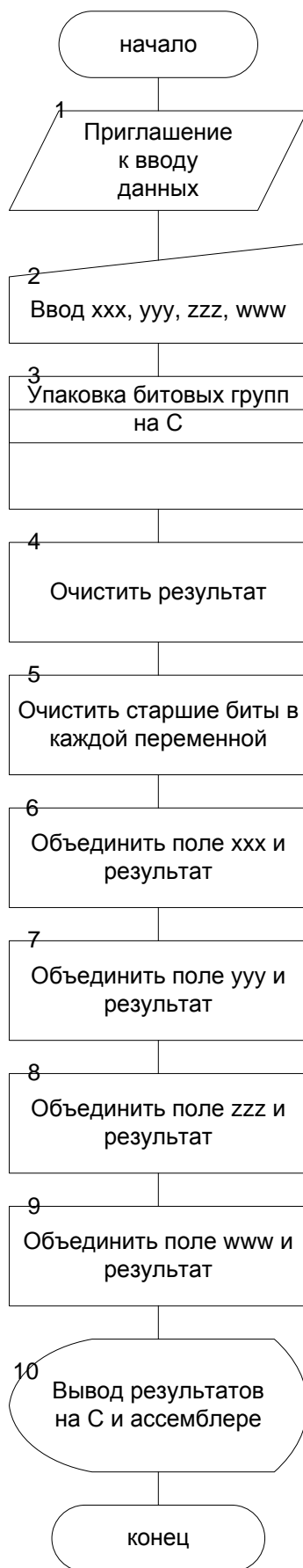
Объединить поле `yyy` и число `pack` на ассемблере;

Объединить поле `zzz` и число `pack` на ассемблере;

Объединить поле `www` и число `pack` на ассемблере;

Вывести значения переменных на `C`;

Вывести значения переменных на ассемблере.



## 5. Текст программы с комментариями.

```
//+=====
// File pack.cc
// Упаковка битовых групп
//
// Эта программа упаковывает битовые группы в длинное беззнаковое число
//
// (С) Дужий В.И., 2012
//
// Входные данные:
//          7----5-----0
//          ! ?? !   xxx !   xxx
//          +----+-----+
//          6
//          15---8-----0
//          ! ?? !   yyy !   yyy
//          +----+-----+
//          9
//          15---11-----0
//
//          ! ?? !   zzz !   zzz
//          +----+-----+
//          12
//          7----4-----0
//          ! ?? !   www !   www
//          +----+-----+
//          5
// Выходные данные:
// Длинное целое беззнаковое число, которое содержит указанные битовые группы
//
//          31----+-----+-----+-----0
//          ! xxx ! yyy ! zzz ! www !   value
//          +----+-----+-----+-----+
//          6      9      12      5
//
//-----
#include <iostream>
#include <iomanip>
using namespace std;

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

int main()
{
    cout<<"\n\t\t(С) Дужий В.И., 2012"
         <<"\n\t\tРаспаковка битовых групп"
         << "\n\tУпаковать битовые группы, содержащиеся в целых числах,"
         << "\n\t\tв 32-битовое целое число Value";
    for (;;)
    {
        cout << "\nПожалуйста, введите 2 16-ые цифры для XXX (например, 5a):";
        cin  >> hex >> tmp; xxx = tmp;
        cout << "Пожалуйста, введите 3 16-х цифры для YYY (например, 9db):";
        cin  >> yyy;
        cout << "Пожалуйста, введите 4 16-х цифры для ZZZ (например, 8e7f):";
```

```

    cin  >> zzz;
    cout << "Пожалуйста, введите 2 16-х цифры для WWW (например,   14):";
    cin  >> tmp; www = tmp;
// Сделать копии исходных чисел для фрагмента на ассемблере
    xxx_a = xxx; yyy_a = yyy; zzz_a = zzz; www_a = www;
//===== C =====
// Очистить старшие биты в каждом исходном числе
    www &= 0x1f; // очистить все биты в числе, кроме 5 младших (4-0)
    zzz &= 0xfff; // очистить все биты в числе, кроме 12 младших (16-5)
    yyy &= 0x1ff; // очистить все биты в числе, кроме 9 младших (25-17)
    xxx &= 0x3f; // очистить все биты в числе, кроме 6 младших (31-26)
// Объединить каждое битовое поле с результирующим числом Value
    value = xxx; // объединить поле xxx с value
    value = (value << 9) | yyy; // объединить поле yyy с value
    value = (value << 12) | zzz; // объединить поле zzz с value
    value = (value << 5) | www; // объединить поле www с value
//===== Assembler =====
// Упаковать битовые группы
    __asm{
// Очистить старшие биты в каждом исходном числе
        andb    xxx_a,0x3f
        andw    yyy_a,0x1ff
        andw    zzz_a,0xffff
        andb    www_a,0x1f
        xorl    eax,eax
// объединить поле xxx с value
        orb     al,xxx_a
        shll    eax,9
// объединить поле yyy с value
        orw     ax,yyy_a
        shll    eax,12
// объединить поле zzz с value
        orw     ax,zzz_a
        shll    ax,5
// объединить поле www с value
        orb     al,www_a
        movl    value_a,eax
    };
// Форматный вывод результатов
    cout << hex
        << "Результирующее упакованное число (C++): " << value
        << "\nРезультирующее упакованное число (Asm): " << value_a
        << endl;
};
    return 0;
}

```

**Тестовые примеры.**

Номер	Исходные данные				Ожидаемый результат	Полученный результат	Цель теста
	xxx	yyy	zzz	www			
1	3f	1ff	fff	1f	FFFF FFFF		Все биты равны 1
2	2a	155	555	a	AAAA AAAA		Чередующиеся 1 и 0
3	15	aa	aaa	15	5555 5555		Чередующиеся 0 и 1
4	3f	0	0	0	FC000000		Все 1 в битовой группе xxx
5	0	0	0	1f	1F		Все 1 в битовой группе www
6	0	1ff	0	0	3FE0000		Все 1 в битовой группе yyy
7	0	0	fff	0	1FFE0		Все 1 в битовой группе zzz
8	02	11a	2b3	18	1234 5678		Произвольные биты
9	06	115	9e2	0d	1A2B 3C4D		Произвольные биты
10	ea	ff55	f555	ea	AAAA AAAA		Влияние несущественных битов
11	15	aa	aaa	15	5555 5555		Влияние несущественных битов