## Задача 1:

Направете функция, която разменя младшия и старшия байт на променлива от тип **uint16\_t**.

**void swapBytes(uint16\_t\* word);**

## Задача 2:

* Реализирайте функция, която печата битовете на число с плаваща запетая от тип **double.** (прочетете за double precision floating point format)
* Реализирайте функция, която печата стойностите на знаковия бит, експонентата и мантисата на число с плаваща запетая от тип **double.** Стойностите да се изпечатат в десетична бройна система все едно са интерпретирани от двоичен прав код.

## Задача 3:

Реализирайте функции, чрез които да се менажира колекция от хора. Използвайте следната структура:

**struct Person {**

**char name[100];**

**char id[ID\_LEN];**

**?? birthYear ??;**

**?? birthMonth ??;**

**?? birthDay ??;**

**?? gender ??;**

**}**

Изберете подходящи типове за полетата **birthYear, birthMonth, birthDay**, **gender** така че структурата да използва възможно най-малко памет.

Всеки човек има уникален идентификатор, състоящ се от няколко главни или малки латински букви. **ID\_LEN** е препроцесорна константа.

Реализирайте следните функции:

**void\* initArray(size\_t capacity);**

Заделя в **димамичната памет** масив, в който могат да се запазят най-много **capacity** на брой елемента от тип **Person**.

**int addPerson(char \*name, unsigned short birthYear, unsigned char birthDay, char gender);**

Добавя човек към динамичния масив. Избира уникален идентификатор (id) за човека. При успех връща 0, а при неуспех -1.

**int removePerson(char \*id);**

Премахва човек от динамичния масив по подаден идентификатор (id). При успех връща 0, а при неуспех -1.

**void readPerson();**

Прочита даннни за човек от стандартния вход и го добавя към динамичния масив.

**int printPerson(char \*id);**

Печата данни за човек с определен идентификатор (id) на стандартния изход. При успех връща 0, а при неуспех връща -1.