MMS C Camp - Oct-Nov 2021
КОНТРОЛНО №2

ЗАДАЧА 1: - 10 т.

Направете структура **Car**, за която да се съхраняват следните данни:

- model - *C-string*
- maxspeed - *uint8_t*
- price - *double*

Да се направят 6 функции, с помощта на които да се сортира масив от автомобили (сравняващи функции каквито се подават в стандартната **qsort()**):

1. Сравнение по марка - възходящ лексикографски ред;
2. Сравнение по марка - низходящ лексикографски ред;
3. Сравнение по максимална скорост - възходящ ред;
4. Сравнение по максимална скорост - низходящ ред;
5. Сравнение по цена - възходящ/низходящ ред.
6. Сравнение по цена - възходящ/низходящ ред.

Да се направи функция, която да връща сравняваща функция, от тези които дефинирахме по-горе.

int (*getComparator(int n))(const void*, const void*);

Номерът на функцията **n**, да е между 1 и 6. При число различно от това, да се върне нулев указател.

Да се създаде масив от 10 автомобила. Да се запълни с произволни стойности:

1. Марка - **[A-Z][a-z]{4,10}**
2. Максимална скорост - 100 - 300
3. Цена - 1000.00 - 100 000.00

Потребител да въведе от стандартния вход цяло число между 1 и 6, с което да избира начин на сортиране на масива от автомобили, като използва функцията **getComparator().**
Сортирането да се извърши със стандартната функция **qsort().**
Да се изведе сортираният масив в подходящ формат.

MMS C Camp - Oct-Nov 2021
КОНТРОЛНО №2

ЗАДАЧА 2: - 10 т.

Обяснете подравняването на следните структури и обединения. Скицирайте как се помещават в паметта и пресметнете размерът им в байтове (какъвто ще го изчисли **sizeof()**).

Приемете следните размери (в байтове): **char** - 1, **short** - 2, **int** - 4, **long** - 8,**float** - 4, **double** - 8, **pointer** - 8.
*Позволено е да използвате **sizeof()** за да проверите размера на структурата/обединението, но трябва да обосновете резултата.*

a)
```
struct stu_a {
    int i;
    char c;
};
```

b)
```
struct stu_b {
    char* p;
    char c;
};
```

c)
```
struct stu_c {
    int i[2];
    long l;
    char c;
};
```

d)
```
struct stu_d {
    char c1;
    int i[3];
    char c2;
};
```

e)
```
struct stu_c {
    char c;
    struct stu_d;
};
```

f)
```
struct stu_f {
    int a : 2;
    int b : 4;
};
```

g)
```
union un_g {
    uint8_t a;
    uint16_t b
    uint8_t* p;
};
```

h)
```
union un_h {
    uint32_t a;
    struct {
        uint16_t b;
        uint16_t c;
    };
};
```

i)
```
union un_i {
    uint32_t a;
    uint8_t b[4];
};
```

MMS C Camp - Oct-Nov 2021
КОНТРОЛНО №2

ЗАДАЧА 3: - 10т.

Въвежда се до 500 цифрено цяло число. Напишете програма, която кодира числото по следния начин:

- Ако една цифра е на четна позиция, тя се замества със съответстващата главна латинска буква. Например:
  - 0 се замества с A
  - 1 се замества с B
  - 2 се замества с C и т.н.
- Ако една цифра е на нечетна позиция, тя се замества със символ, както е зададено в долната таблица:

Цифра	Символ
0	!
1	#
2	/
3	~
4	=
5	`
6	\
7	>
8	.
9	,

Примерен вход:

10296126782646987676234

Примерен изход:

B!C,G#C\H.C\E\K.H\H\C~E