| – | + | + | + |

Контрольная работа №1.

(Статья 1, УК ПРАК. За хранение и использование запрещённых материалов, а также списывание, вопрос или ответ соседу по теме или нет – комиссия без понижения оценки)

Вариант 1. *ФИО:* Холопкин, 211 гр.

Задание №1.

Написать функцию NUM, которая возвращает **количество включений** второй строки в первой. Стандартными функциями работы со строками пользоваться нельзя.

Примеры:

NUM("Студент любит учиться, учиться и ещё раз учиться", "учиться") == 3,
NUM("Студент МГУ – всем школьникам пример", "") == -1,
NUM("Каждый студент должен знать, что такое NULL", NULL) == 0.

Задание №2.

Только с помощью операторов **'-', '&'**, '~', **'>>'** (каждый можно использовать лишь один раз) и произвольного количества операторов '=', '<<' и '+' реализовать **вычитание с сатурацией (округлением)** (DEC_SAT) для чисел в диапазоне 0..255 (таким образом, циклы и функции использовать нельзя!!!) sizeof(char) == 1.

Пример:

int src0, src1, dst;

dst = DEC_SAT(src0, src1);

src0	src1	dst
20	-15	-15
0	20	0
-132	-222	-222

Задание №3.

Допустимо ли в Си? Если «да» – **опишите** семантику каждого правильного действия, состояние массива и указателей после каждого их изменения. Если есть ошибка – **объяснить**, в чем она состоит. Далее считать ошибочную строку вычеркнутой из текста программы. sizeof(int) = 4; sizeof(char) = 1. **Что будет напечатано?** *(давать ответ можно прямо на задании)*

(+)

```
int   a[5] = {0,1,2,3,4};
int*  pa = NULL;
int*  pb = NULL;
char* pc = NULL;
int   i;

✓ pa = a;  – pa указ. на 0 эл-т массива a
✓ pa = &a[0];  – аналогично
✓ pc = (char*)(pa + 3); – pc указ. на 3 эл-т мас. как на char
✓ pa[2] = 5;  – во второй эл-т массива запишется 5
err a[3] = 1 & (int)(pa + pb);  – ошибка: нельзя складывать указатели
✓ pb = &4[a];  – pb – указ. на 4 эл-т массива
✓ *pc += pa[4] - pb[0];  – pa[4] - pb[0] = 4 - 4 = 0 => байт, на кот. указ. pc - не изм.
✓ pb = pa + 1; – pb указ. на 1 эл-т массива
✓ *(a+5) = 11;  – выход за пределы массива
✓ pc[-12] = (pb - pa) & 4; – в самом 1-м байте массива остается 0, т.к. (pb-pa)&4 = 1&4 = 0
✓ pb[1] = *(2+pa); – во 2-й эл-т массива запишется он сам

for (i=0;i<5;i++)
  printf("%d ", a[i]); printf("\n");
```

Напечатано:
0 1 5 3 4

| 0 | 1 | 5 | 3 | 4 | – сост. массива

Задание №4.

Написать функцию, которая по двумерному целому массиву размерности **nxm** строит новый массив размерности **nx1** следующим образом: элемент массива-результата определяется как значение максимального элемента соответствующей строки двумерного массива. Размерности массива «**n**» и «**m**» должны передаваться в функцию в качестве параметров.

Дополнительные требования.

Для решения задачи НЕЛЬЗЯ использовать:

1) *циклы for;*
2) *циклы while;*
3) *goto;*
4) *рекурсию.*

Вариант 1. **ФИО:** Холопкин, 211 гр.

Задание 2.

dst = src0 - src1;
dst = dst & (~(dst >> 31))

если dst < 0, то dst >> 31 = 11...1;
~(dst >> 31) = 00...0 => ответ 0;
если dst ≥ 0, то dst >> 31 = 00...0;
~(dst >> 31) = 11...1 => ответ dst

(+)

Задание 1.

```
int num (const char *str1, const char *str2)
{
    int i=0, j=0;
    int count=0;
    if ((str2 == NULL) || (str1 == NULL))
        return 0;
    if (str2 == '\0')
        return -1;
    while (str1[i] != '\0')
    {
        if (str1[i] == str2[j])   // Это while?!
            j++;
        if (str2[j] == '\0')
        {
            j=0;
            count++;
        }
        i++;
    }
    return count;
}
```

(—)

Работает только при strlen(str2)==1

str1 str2
("aab", "ab")

Задание 4.

```
int *createarr( int **mas, int n, int m)
{
    int *trash = (int*) malloc (n*sizeof(int));
    int i=0;
    int j, max;

    do
    {
max = mas [i][0] ; j=1;
if (j != m)
    do
    {  if (mas[i][j] > max)
           max = mas[i][j];
       j++;
    }
while (j<m);
trash[i] = max;
i++;
}
while (i<n);
return trash;
}
```

лучше не надо

MAS[i*n+0]

Компилятор не умеет про размер строки

V! Критичная ошибка

+

// этими 2-мя строками - // отступ влево