

+ | + | + | +

# 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: Белишев М.В. 211ф

### 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. (То есть циклы и функции использовать нельзя!!!) sizeof(char) = 1.

Примеры:  
int src0, src1, dst;  
dst = DEC\_SAT(src0, src1);

src0	src1	dst (результат)
20	15	5
0	20	0
132	222	0

### Задание №3.

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

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

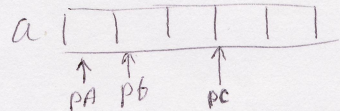
```
+ pa = a; // pa на 1-й элемент массива a;
+ pa = &a[0]; // pa на первый элемент массива a
- pc = (char*) (pa + 3); // pc на 3-й элемент массива, как char.
+ pa[2] = 5; // a[2] = 5, pa остается на a[0]
+ a[3] = 1 & (int)(pa + pb); // ошибка: нельзя сложить адреса
+ pb = &a[4]; // pb на 4-й элемент массива
+ *pc += pa[4] - pb[0]; // значение char, на которое указывает pc не ищет. т.к. 4-4=0
+ pb = pa + 1; // pb указывает на a[1]
+ *(a+5) = 11; // выход за пределы массива a[5] = 11
- pc[-12] = (pb - pa) & 4; // указатель на 1-ый элемент массива, т.к. адреса int имеют 4 байта
+ pb[1] = *(2+pa); // a[2] = 5
```

можно выйти за массив? нет.

Какой?

+

0 1 5 3 4 11  
~~0 1 5 3 4 1~~  
0 1 5 3 1 1

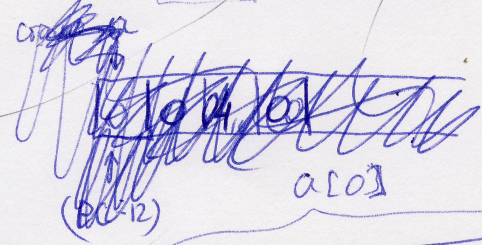
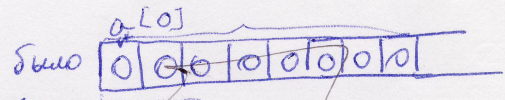


### Задание №4.

Написать функцию, которая по двумерному целому массиву размерности  $n \times m$  строит новый массив размерности  $n \times l$  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) цикл for();
  - 2) цикл while();
  - 3) goto;
  - 4) рекурсию.

pc[-12] = 4 & 4; pc[-12] = 4;



Бред

стало: 4 0 0 0 0 0 0  
пусть сначала младш. разряды потом старшие.

Вариант 1.

ФИО: Бельшинов М.В

51  
int NUM (char \*str, char \*etalon)

{  
int count=0, l=0, k=0;  
if ((\*etalon) == '\\0') {  
return -1;  
}

if (l == strlen(etalon)) {  
return 0;  
}

while (str[l] != '\\0') {

Здесь? if ((str[l] == (\*etalon)) {

while ((etalon[k] != '\\0') && (str[l+k] == etalon[k])) {  
k++;

}  
if ((etalon[k] == '\\0')) {

count++;  
l = l+k-1;

}

}

l++;  
k=0;

}

return count;

}

(He mago)  
~~if (str[l+k] != '\\0')~~

(+)

24

int \*mass (int n, int m, int \*arr)

int \*helper, max, l=0, k=0;

~~helper = arr;~~  
if (sizeof(arr) >= 20) {  
return NULL;  
}

Belega sizeof (int \*) → 4  
→ 8

helper = malloc (sizeof(int) \* n);

do {

~~max = arr[l][k];~~

arr - ymagazens na int

arr [ ] → int, a b [ ] - omdaka kommutatsiya

do {

~~max = helper[l][k];~~

~~if (helper[l]~~

if (arr[l][k+1] > max) {

max = arr[l][k+1];

}

k++;

Esli m == 1, to upredelennye

} while (k < m);

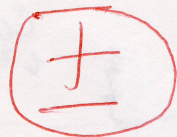
helper[l] = max;

l++;

k=0;

} while (l < n);

return helper;

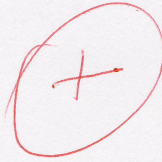


Вариант 1.

ФИО: Белышев М В 211 ф

№2

$dst = src0 - src1;$   
 $dst = [ \sim (dst \gg 8) ] \& dst;$



если  $src0 - src1 < 0$

$dst = src0 - src1$  ~~гас~~

1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---

$dst \gg 8$  гас 

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

$\sim (dst \gg 8)$  гас 

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

если  $src0 - src1 \geq 0$

~~$dst$  гас~~

$\sim (dst \gg 8)$  гас 

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

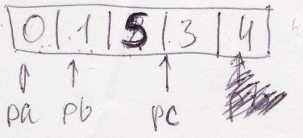
в итоге, если ~~src~~

~~dst~~ разность отриц. то битовое 'и'

гас нуль, иначе битовое 'и' гас

каждой аргумент, т.е.  $dst$

Белышев



Мет. ст.  
ст. м.

4 1 5 3 4

$(= & \sim \gg) \text{ 1 раз}$   
 $= \ll + \text{ -мощно}$

$\sim [(src0 - src1) \gg 8] \&$

$dst = (src0 - src1)$

$[\sim (dst \gg 8)] \& dst$

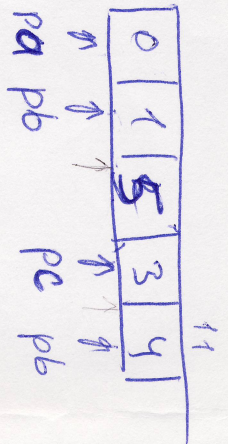
# Черновик

~~а~~ b

a = 3  
b = 5

b =  $(\sim a \gg 8) \& b$

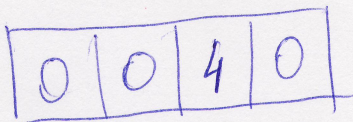
0 1 5 3 1 1



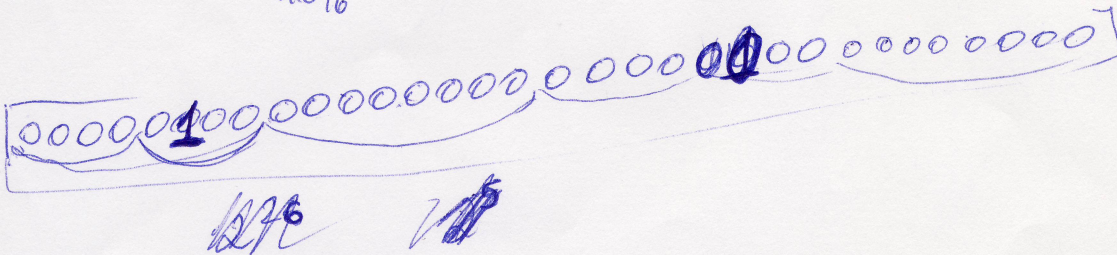
5 - 3

t1 = 0

t = 1 - j  
t1 = t >> 32  
min = j + t & t1



$40_{16} = 16 \cdot 4 = 64$

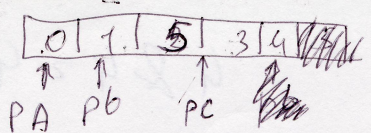


4 & 4 = 4 ?

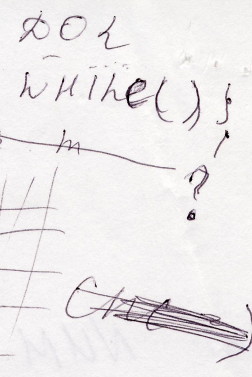
# Черновик

```
int NUM (char *str, char *etalon)
1) {
    int count, i = 0;
    int k = 0, j = 0;
    if ((*etalon) == '\0') {
        return -1;
    }
    if ((etalon) == NULL) {
        return 0;
    }
    while (str[i] != '\0') {
        if ((str[i]) == (*etalon)) {
            while ((etalon[k] != '\0') && (str[j] == etalon[k])) {
                k++;
                j++;
            }
            if ((str etalon[k] == '\0')) {
                count++;
            }
        }
        i++;
    }
    return count;
}
```

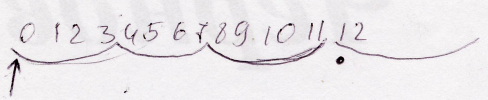
abcdef      cd  
0 1 2 3 4 5      0 1



$$PC[12] = (pb - pa) \& 4$$



**\*\*ent**



... 100,

```

int *mass (int n, int m, int *arr) {
  int *helper;
  if (sizeof(arr) != 0) {
    return NULL;
  }

```

$$a[5] = 0, 1, 2, 3, 4$$

```

  helper = malloc (sizeof(int) * n);

```

$$*(a+5) = 11$$

```

max
max = helper

```

```

  do {
    max = helper[0][k];
    if (helper[0][k+1] > max) {
      max = helper[0][k+1];
    }
    k++;
  } while (k > m);

```

```

k = 0;
l++;
max = 1
do {
  do {
    max1 = helper[l][k];
    if (helper[l][k+1] > max1) {
      max1 = helper[l][k+1];
    }
    k++;
  } while (k > m);
  k = 0;
  l++;
  if (max > max1) {
    max = max1;
  }

```