Контрольная работа №1.

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

Вариант 2. ФИО: Кортжов А.А. 211 гр.

Задание №1.

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)*

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

pa = &a[0];
pa = a;
pc = (int*)(4 + a);
1[pa] = 3;
a[0] = 0 & (int)((pa++) + pc);
pc = a[4];
pb = pa + 1;
(pc-1)[0] += pa[3] - pb[0];
*(a+3) = 11;
pb[-1] = pb - pa;
for(i=0;i<5;i++) printf("%d ", a[i]); printf("\n");
```

Задание №2.

Римский император Цезарь для шифрования своих писем использовал такой алгоритм: каждому дню в году соответствовало некоторое число (КОД), и каждая буква (исключая знаки препинания и числа) в секретной депеше заменялась на символ, стоящий на позиции на КОД правее первоначального. Если номер «выходит» за рамки алфавита, то отсчёт начинается заново.

Составьте программу для латинского алфавита, облегчающую работу доверенному шифровальщику Цезаря.

<u>Входные данные</u>.
С консоли вводится КОД N (где 0 <= N <= 255) и строка сообщения (не более 254 символов).
Например:

1
Victory

<u>Выходные данные</u>.
На экран выводится зашифрованное сообщение.
В данном случае:

Wjdupsz

<u>Дополнительные требования</u>.
Для решения задачи НЕЛЬЗЯ использовать:
1) цикл for,
2) цикл do…while,
3) goto;
4) рекурсию.

Задание №3.

Написать функцию. Через параметр в функцию подаётся строка произвольной длины. Слова в ней разделены произвольным количеством пробелов. Необходимо создать новую строку, в которую надо скопировать через ОДИН пробел все слова исходной строки, упорядоченные в порядке убывания их длины. Передать полученную строку в качестве возвращаемого значения функции. *Дополнительное требование: НЕЛЬЗЯ использовать стандартные функции работы со строками (то есть strlen, ит.п..)*

Задание №4.

Найти **минимум** (MIN) из двух int чисел, если известно, что **каждое из них меньше 2^{29}**. Реализовать только с помощью операторов **'+', '&', '>>'** (каждый можно использовать ТОЛЬКО один раз) и произвольного количества операторов '=' и '-'. (То есть, циклы и функции использовать нельзя!!!) sizeof(int) = 4.

Примеры:

```
int        src0, src1, dst;
dst = MIN(src0, src1);
```

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

Вариант 2.   ФИО: ____________________

№4

```
int MIN(int a, int b)
{   int tmp0, tmp1;
    tmp0=a-b;
    tmp1 = tmp0 >> 1000;  // tmp1==0, если a≥b
                          // tmp1==-1, если a<b.
  tmp1= b + (tmp0&tmp1); // tmp1==b, если a≥b
                         // tmp1==b+a-b, если a<b.
  return tmp1;
}
```

Обязательны

±

№2

```
#include <stdio.h>
#define SIZE 26
int n;
void convert(char c, char reg) // reg=='z', если меньше букв и 'Z' иначе.
{
  char trash;
  trash= c+(char)n; // сдвигаем
  while (trash>reg)
        trash-= SIZE;  //передвигаем символ в „зону букв" a...z  A...Z
  printf("%c", trash);

}
```

+
−

```
void main()
{
  char str[];
  char c;
  int n;
  scanf("%d", &n);
  scanf("%s", str);
  while (*str!='\0')
  {
     if ((*str>='a')&&(*str<='z'))
        convert(*str, 'z');
     else
        if ((*str>='A')&&(*str<='Z'))
            convert(*str, 'Z');
        else
            printf("%c", *str);
   str++;
  }
}
```

Размер обязателен!

V

Вариант 2. ФИО: ____________________

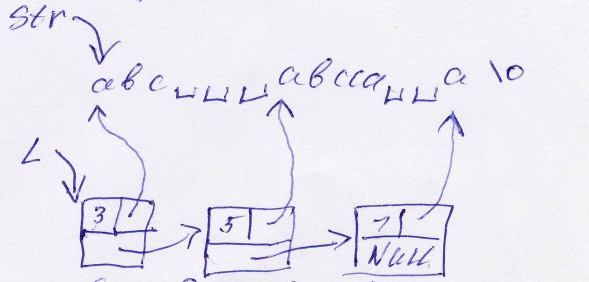
№3

```
typedef struct list_struct.
{
    int length;
    char* start;
    (struct list_struct) *next;
} list;

char* sort (char* str)
{
  List L;
  int length=1; // для '\0'; для calloc.
  int len ; // для каждого слова.
  while (*str!='\0')
   {
       if (*str!='_')   space
       {
```

нет. идея не для контрольной.
много писать

идея:



вставка в L будет сразу сортироваться

```
readword(char *str,                    )
{

   while ((*str!='\0')&&(*str!='_'))   space
     {
```

Жаль.

Черновик

1 2 3 4 ✓ ?

~~int Mi~~

```
int min(int a, int b)
{
tmp0 = a-b;
tmp1 = tmp0 >> много;
tmp1 = b + tmp0 & tmp1.
return
```

(1)

1 —— a<b

0 —— a≥b

tmp1

111 000

a<b a≥b

f(a<b)

a-b+t

```
#include <stdio.h>
#define SIZE 26 ?

void main()
{
char str[]; char c;
int n;
~~scanf(&n)~~
scanf("%d",&n);  (?)
scanf("%s", str);
while (*str != '\0')
{
c = *str;
сдвиг
```

да

```
do
{
}
while();
```

A ... Z a ... z

26?

сдвиг:

~~a=c+(char)n;~~

~~if (a>c~~

```
if ((c>='a') && (c<='z')) { A }
else if ((c>='A') && (c<='Z')) { B };
```

```
A: a=c+(char)n;
while (a>'a'+SIZE)
a -= SIZE;
printf("%c",a);
```

void convert(char)

```
"calloc(length, char);
                +1 '\0'

char*  sort(char* str){
{
  int Length=1; //общая
  int. Len;     //для кажд. слова

  while (*str!='\0')
  {
    ~~Length++~~, пробел
    ~~if (*str!=' ')~~
    if(*str!=' ')
    {
      Len.

    pa=ka

    }
    ~~Length+=Len;~~
    Length+=(Len+1); //длина+пробел
  str++;
  }

}
```

// считаем кол-во символов без пробелов и сразу формируем структ

|abc␣␣␣abcccc␣␣c|

```
~~struct List_str~~

typedef struct List_struct
{
  int Length;
  char* start;
  (struct List_struct) *next;
} List;

List L;

void. readword(char*str, )
{
  while
```

a[4] = *(a+4);