1 БИЛЕТ(+):

Цезарь на НАМ

(для входного алфавита - русского, для английского аналогично)

\*а->г\*

\*б->д\*

\*в->е\*

\*г->ё\*

\*д->ж\*

\*е->з\*

\*ё->и\*

\*ж->й\*

\*з->к\*

\*и->л\*

\*й->м\*

\*к->н\*

\*л->о\*

\*м->п\*

\*н->р\*

\*о->с\*

\*п->т\*

\*р->у\*

\*с->ф\*

\*т->х\*

\*у->ц\*

\*ф->ч\*

\*х->ш\*

\*ц->щ\*

\*ч->ъ\*

\*ш->ы\*

\*щ->ь\*

\*ъ->э\*

\*ы->ю\*

\*ь->я\*

\*э->а\*

\*ю->б\*

\*я->в\*

\*->.

->\*

Билет 2(+):  
Проверка палиндрома числа на Си

двоичное число

#include <stdio.h>

#include <limits.h>

int main()

{

int a, b = 0, flag = 1;

scanf("%d", &a);

for (int a\_test = a; a\_test != 0;)

{

if (INT\_MAX/10 < b) { printf("Perepolneny\n"); flag = 0;  break;}

else

{

b = b \* 10 + a\_test % 10;

a\_test /= 10;

}

}

if (flag == 1)

{

if (a == b) printf("Polindrom\n");

else printf("NonPolindrom\n");

}

system("pause");

return 0;

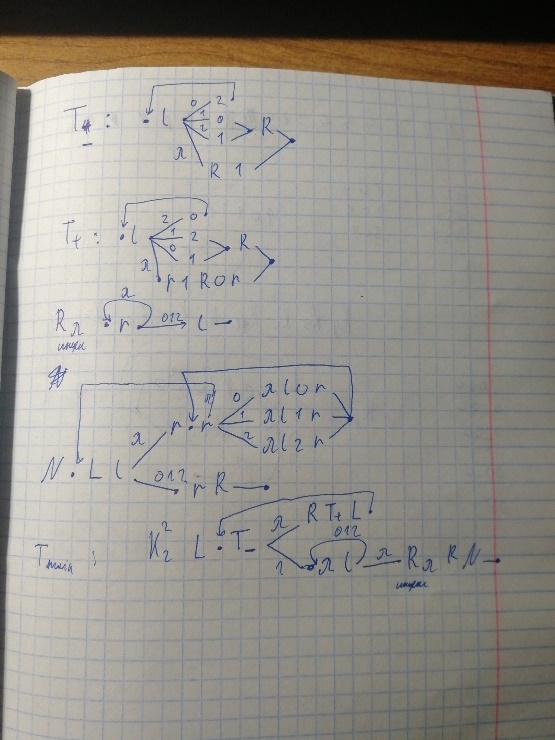
}

Билет 3(+):

Целочисленное деление в кардинальной сс на Си. (НА ДТ)

[#include](https://vk.com/im?sel=134848551&st=%23include) <stdio.h>   
  
int sep(char c){   
if (c == ' ' || c == '\t' || c == '\n')   
return 1;   
return 0;   
}   
  
void write(int a, int b){   
a--; b--;   
if (b == 0)   
printf("Division by zero\n");   
else{   
printf("%d/%d = %d\n", a, b, a/b);   
for(int i = 1; i <= (a/b)+1; i++)   
printf("|");   
printf("\n");   
}   
}   
  
int main(){   
char c = '0', prec = '0';   
int a = 0, b = 0;   
  
while(scanf("%c", &c)){   
if (sep(c)){   
if (sep(prec))   
continue;   
if (a != 0){   
write(a, b);   
b = 0;   
}   
a = b;   
b = 0;   
}   
else if (c == '|')   
b += 1;   
prec = c;   
}   
return 0;   
}

Билет 4(+):  
**Сложение в троичной системе счисления на ДТ**



Билет 5(+):  
**Вывести первые 500 простых чисел на Си**

#include <stdio.h>

int main(){

    int i=2, k, m, n=500;

    while (n>0){

            m=i;

            k=0;

            while (m>1){

                    if (i%m==0) k=k+1;

                    if (k>1) break;

                    m=m-1;

            }

            if (k==1) {

printf ("%d ",i);

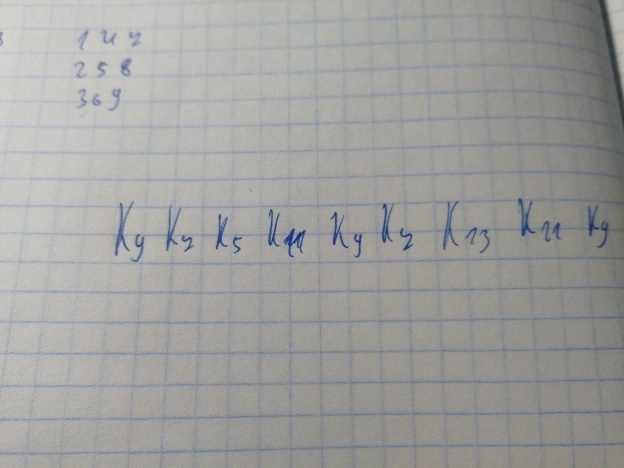
                                             n=n-1;

}

            i=i+1;

    }

}

Билет 6(???):  
Транспонирование матрицы 3 на 3  


Билет 7(+):  
**Реверс байтов целого числа на Си**

#include <stdio.h>

int main()

{

int num1, num2;

printf("Введите исходное целое число ");

scanf("%d", &num1);

// вариант со сдвигом

num2 = ((num1-(num1>>8))<<8+(num1>>8));

printf("Результат реверса байтов %d", num2);

return 0;

}

Билет 8(---):  
**Поразрядная конъюнкция двоичных чисел на ДТ**

**Билет 9(+):**

**НАМ проверка лексикографического упорядочения двоичных слов**

на вход поступают двоичные числа равной длины через символ \_

\_ -> \-/

1\ -> \1

0\ -> \0

\0<0 -> \

\0<1 -> YES

\1<1 -> \

\1<0 -> NO

0<0 -> <00

1<0 -> <01

0<1 -> <10

1<1 -> <11

YES0 -> YES

YES1 -> YES

YES-/ -> YES

NO0 -> NO

NO1 -> NO

NO-/ -> NO

-/0 -> <0-/

-/1 -> <1-/

\-/ -> EQUAL

Билет 10(---):  
**ДТ проверки делимости на 3**

Билет 11(+):  
**НАМ: умножение двух чисел в натуральной системе счисления**

решение, м? :) 1111#11111 через решетку два t числа

|ab->ab|b

ab->

|b->b|

#||->ab#|

#|->

b->|

Билет 12(+):  
**написать Дифференциал многочлена на си  
инпут:2 и 4 3 2   
аутпут:8 3 0**

//Дифференциал многочлена (на вход поступают

//коэффициенты многочлена, на выход - коэффициенты

//производной от многочлена)

#include <stdio.h>

int main(){

   int a[100];

   int power = -1;

   while(power > 100 || power < 0){

   printf("Введите степень многочлена:\n");

   scanf("%d",&power);

   }

   printf("Введите коэффициенты: \n");

   for(int i = 0; i <= power; i++)

   scanf("%d",&a[i]);

   printf("Результат:\n");

   for(int i = 0; i < power; i++)

   printf("%d ", a[i]\*(power-i));

   printf("0\n");

}

Билет 13(+):  
**Проверка палиндрома с помощью НАМ**

либо двоичное число

двоичное число

решение, м? :) всего три буквы, а то алфавит очень большой, много кода. Тесты: abba ababa abbc bacb bbb cab.

##a->a##

##b->b##

##c->c##

a(a)->(a)a

b(a)->(a)b

с(a)->(a)c

a(b)->(b)a

b(b)->(b)b

с(b)->(b)c

a(c)->(c)a

b(c)->(c)b

с(c)->(c)c

#(a)a->#

#(b)b->#

#(c)c->#

#(a)#->##

#(b)#->##

#(c)#->##

a#->(a)#

b#->(b)#

c#->(c)#

#-0->-0

)-0->-0

(-0->-0

a-0->-0

b-0->-0

c-0->-0

1###->.1

##->-0

1-0->.0

->1###

Билет 14(+):  
**Двоичный реверс машинного слова на Си**

Предположим, на вход подаётся целое число в десятичной системе счисления, целого типа с длиной 32 бита. Тогда реверс двоичного слова будет выглядеть так:

#include <stdio.h>

int a, b;

int main()

{

   scanf("%d",&a);

   b=0;

   while (b<32){

       printf("%d", ((a>>b)&1));

       b++;

   }

   return(0);

}

Билет 15(-???):  
**проверка на палиндромию машинного слова на си**

**Написано из предположения что на вход поступает целое число типа int.**

#include <stdio.h>

int a, b, c;

int main()

{

   scanf("%d",&a);

   b=0;

   c=1;

   while (b<16){

       if (((a>>b)&1)&(a>>(31-b)&1)==0) c=0;

       b++;

   }

   if (c==0) printf ("NOT PALINDROM\n");

   if (c==1) printf ("PALINDROM\n");

   return 0;

}

Билет 16(+):

**написать нам переводящий число в двоичной системе счисления в восьмеричную**

Решение:

a1->1a

a0->0a

a->b

000b->b0

001b->b1

010b->b2

011b->b3

100b->b4

101b->b5

110b->b6

111b->b7

10b->b2

11b->b3

01b->b1

1b->b1

0b->b

b->.

->a

Билет 17(---):  
**Составить ДТ по сложению двоичных чисел**

Билет 18(---):  
**ДТ: перевод числа из двоичной системы в натуральную**

**Билет 19(---):  
Написать НАМ построения машинного слова по заданной двоичной маске**

**Билет 20(+):  
Выдача байта на си**

[#include](https://vk.com/im?sel=154279619&st=%23include) <stdio.h>   
  
int main(){   
  
int number,byte;   
printf("введите число и номер байта считая справа\n");   
scanf("%d %d",&number,&byte);   
while (byte < 1 || byte > 4) {   
printf("байт некорректно введен\n");   
scanf("%d",&byte);   
}   
number = number » (byte - 1) \* 8;   
for (int i = 7; i >= 0; —i) {   
printf("%d", (number » i) & 1u);   
}   
printf("\n");   
return 0;   
}

**Билет 21(+):  
НАМ реверса слова**

(A и В - любые (не обязательно различные) символы исходного слова отличные от $ ? #)

?|->.

?A -> >A

?B -> >B

>AA -> A>A

>AB -> B>A

>BA -> A>B

>BB -> B>B

>A| -> <|A

>B| -> <|B

>A -> <|A

>B -> <|B

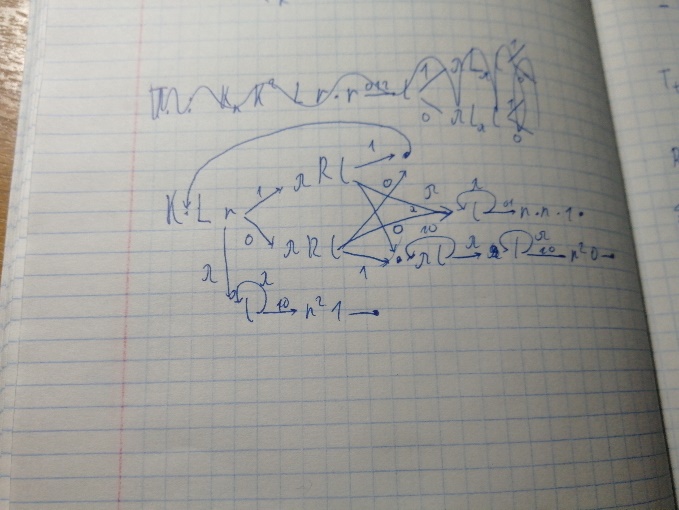
A< -> <A

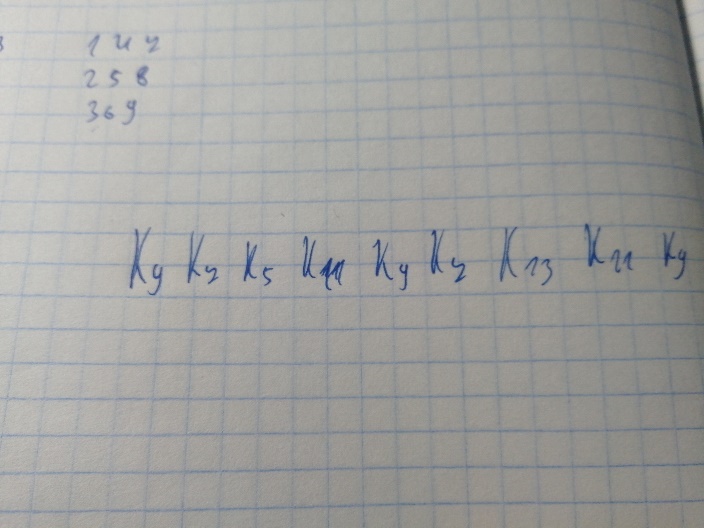
B< -> <B

< -> ?

->?

**Билет 22(-):  
Составить ДТ проверсчки палиндромии слова**



**Билет 23(+?):**

**Билет 25(----):  
Умножение многочленов на Си**

**Билет 26(+):  
обратный код 16 числа нам**

\*0->F\*

\*1->E\*

\*2->D\*

\*3->C\*

\*4->B\*

\*5->A\*

\*6->9\*

\*7->8\*

\*8->7\*

\*9->6\*

\*A->5\*

\*B->4\*

\*C->3\*

\*D->2\*

\*E->1\*

\*F->0\*

\*->.

->\*

**Билет 27(+):  
НАМ сложение двух двоичных чисел**

0#->#1   
1#->0   
#->\*   
\*0->\*   
\*1->\*   
\*+&->.   
&0->0&   
&1->1&   
&->@   
0@->1   
1@->@0   
+@->+1   
+->#+&

**Билет 28(-):  
Си: лексикографическая упорядоченность последовательности слов**

БЕЗ ПРОВЕРКИ СЛОВ, НАЧИНАЮЩИХСЯ НА ОДНУ И ТУ ЖЕ БУКВУ

#include <stdio.h>

#include <ctype.h>

#define OUT 0

#define IN 1

unsigned int char\_to\_set(char c) {

    c=tolower(c);

    if (c< 'a'||c>'z')

            return 0;

    else return 1u<<(c - 'a');

}

int main () {

    int alpha=1;

    int c, state, sim;

    state = OUT;

    unsigned int min = char\_to\_set('a');

    while ((c=getchar())!=EOF){

            sim=char\_to\_set(c);

            if (state==OUT&&sim!=0){

                    if(sim>=min) min = sim;

                    else {alpha=0;break;}

                    state = IN;

            }

            if (state==IN&&(c=='\n'||c=='\t'||c==' '||c==',')) state=OUT;

    }

    if (alpha==1) printf("Упорядочена");

    if (alpha==0) printf("НЕ упорядочена");

}

**Билет 29(+):  
Инкремент 16-чного числа на НАМ**

\*0 -> 0\*

\*1 -> 1\*

\*2 -> 2\*

\*3 -> 3\*

\*4 -> 4\*

\*5 -> 5\*

\*6 -> 6\*

\*7 -> 7\*

\*8 -> 8\*

\*9 -> 9\*

\*a -> a\*

\*b -> b\*

\*c -> c\*

\*d -> d\*

\*e -> e\*

\*f -> f\*

\* -> @

0@ -> 1!

1@ -> 2!

2@ -> 3!

3@ -> 4!

4@ -> 5!

5@ -> 6!

6@ -> 7!

7@ -> 8!

8@ -> 9!

9@ -> a!

a@ -> b!

b@ -> c!

c@ -> d!

d@ -> e!

e@ -> f!

f@ -> @0

@ -> 1!

! ->.

-> \*

**Билет 30(-):  
написать на си конвертацию тьюринговской машины из пятерок в четверки**

Компилировать через gcc, на предупреждения внимание не обращать.

#include <stdio.h>

int main() {

int q0, q1;

char a, b, move;

while (scanf("%d,%c,%c,%c,%d", &q0, &a, &b, &move, &q1)) {

if ((int)move == (int)'u') {

printf("(%d,%c,%c,%d)\n", q0, a, b, q1);

}

else if ((int)a == (int)b) {

printf("(%d,%c,%c,%d)\n", q0, a, move, q1);

}

else {

printf("(%d,%c,%c,%d)\n", q0, a, b, q0 + 100);

printf("(%d,%c,%c,%d)\n", q0 + 100, b, move, q1);

}

if ((int)move == (int)'s') {

break;

}

}

}

**Билет 31(-):  
Написать программу, выводящую график функции на Си**

**#include <stdio.h>  
#include <math.h>  
  
#define WIGTH\_FIELD 20 // ширина поля  
  
int main() {  
    int span = 0; // отступ  
    double real = 0.0;  
    for (double i = 0.0; i < 100; i += 0.4) {  
        real = (sin(i) > 0 ? 1 + sin(i) : 1 - (-sin(i))) / 2.0;  
        if (real == 0.0) {  
            for (int j = 0; j < WIGTH\_FIELD / 2; ++j)  
                printf(" ");  
            printf("\*\n");  
        } else {  
            span = WIGTH\_FIELD \* real;  
            for (int j = 0; j < span; ++j)  
                printf(" ");  
            printf("\*\n");  
        }  
    }  
    return 0;  
}**

**Билет 32(-):**[#include](https://vk.com/im?sel=162450565&st=%23include) <stdio.h>  
  
int function(int mas[100],int n, int x){  
int i, j, z, s;  
for (i=0;i<=n;i++){  
if(i!=0){  
s=x;  
for(j=1;j<=i;j++){  
s=s\*s;  
}  
}else{  
s=1;  
}  
z=z+x\*mas[i];  
}  
return z;  
}  
  
int main(){  
int i,n, funct[100], proiz[100], min, max;  
scanf("%d",&n);  
scanf("%d",&min); scanf("%d",&max);  
for(i=n-1;i>=0;i--){  
scanf("%d",&funct[i]);  
if(i>1) scanf("%d",&proiz[i-1]);  
}  
printf("%d, %d, %d\n",n,min,max);  
for(i=min;i<=max;i++){  
printf("x=%d, funct=%d\n",i,function(funct,n-1,i));  
}  
printf("\n");  
for(i=min;i<=max;i++){  
printf("x=%d\n, proiz=%d\n",i,function(funct,n-2,i));  
}  
return 0;  
}