

[illegible]

```

        {0,0,0,0,0,0,0,0},
        {0,0,0,0,0,0,0,0},
};

int64_t max_custom(int64_t first_num,int64_t second_num){
    if(first_num<second_num)
        return second_num;
    else
        return first_num;
}

void tst_max_castom(){
    int64_t a=0, b=0;
    assert(max_custom(a,b)==0);
    a=999, b=128123;
    assert(max_custom(a,b)==128123);
    a=INT_MAX,b=INT_MIN;
    assert(max_custom(a,b)==INT_MAX);
    a=LLONG_MIN,b=LLONG_MAX;
    assert(max_custom(a,b)==LLONG_MAX);
}

int64_t mlem(int64_t n){
    int64_t smeshenie_po_x=0;
    int64_t mx_elem=LLONG_MIN;
    for(int64_t y=0;y<n;y++){
        smeshenie_po_x++;
        for(int64_t x=0;x<smeshenie_po_x;x++){
            mx_elem =max_custom(mx_elem,matrix[x][y]);
        }
    }
    return mx_elem;
}

void test_mlem(){
    int64_t n=1,a=1;
    for(int64_t i=0;i<n;i++){
        for(int64_t j=0;j<n;j++){
            matrix[j][i] =a;
            a++;
        }
    }
    assert(mlem(n)==1);
    n=2,a=1;
    for(int64_t i=0;i<n;i++){
        for(int64_t j=0;j<n;j++){
            matrix[j][i] =a;

```

```

        a++;
    }
}
assert(mlem(n)==4);
n=2,a=4;
for(int64_t i=0;i<n;i++){
    for(int64_t j=n-1;j>=0;j--){
        matrix[j][i] =a;
        a--;
    }
}
assert(mlem(n)==3);

}

void ymnogenie(int64_t maxelement, int64_t n){
    int64_t smeshenie_po_y=0;
    int64_t sum=0;
    for( int64_t x=0;x<n;x++){
        smeshenie_po_y++;
        for( int64_t y=0;y<smeshenie_po_y;y++){
            matrix[x][y]*=maxelement;

        }
    }
}

}

int main()
{
    tst_max_castom();
    //test_mlem();
    int64_t number=0;
    printf("Enter the number of the tests\n");
    scanf("%lld",&number);
    while(number>0){
        number--;

        int64_t n=0;
        int64_t maxelemet=LLONG_MIN;
        printf("Enter the dimension of the matrix\n");
        scanf("%lld",&n);
        if(n==0){
            return 0;
        }
        printf("Enter the elements of the matrix\n");
        for(int64_t i=0;i<n;i++){
            for(int64_t j=0;j<n;j++){
                scanf("%lld",&matrix[j][i]);
            }

```

```

    }
    maxelemet=mlem(n);
    ymnogenie(maxelemet,n);
    printf("ANSWER\n");

    for(int64_t i=0;i<n;i++){
        for(int64_t j=0;j<n;j++){
            printf("%lld ",matrix[j][i]);
        }
        printf("\n");
    }
}

return 0;
}

```

9. Дневник отладки должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Вре мя	Событие	Действие по исправлению	Примечание

10. Замечания автора по существу работы
Замечания отсутствуют, работа конструктивная.

11. Выводы
От лабораторной работы получил исключительно положительные эмоции и впечатления. По моему мнению, знания, приобретенные мною на данной лабораторной работе, помогли мне лучше осознать принципы работы операционных систем, принципы представления матриц в ЯП Си, принципы их обхода и работы с ними в Си, принцип работы отладчика Си, более пристально смог изучить язык программирования Си, научиться отлаживать свой код и находить ошибки разного типа, что несомненно поможет мне при решении практических задач.

Недочёты при выполнении задания могут быть устранены следующим образом: --

Подпись студента _____