

Задание 5

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    //создаем массив из 9 элементов с индексами(0..8)
```

```
int mas[9]; mas[0] = 99; mas[1] = 88; mas[2] = 77; mas[3] = 66; mas[4] = 55; mas[5] = 44; mas[6] = 33; mas[7] = 22; mas[8] = 11;
```

```
    //вводим целочисленные переменные i и j
```

```
int i, j ;
```

```
    //пока значение i не больше 7 мы выписываем элемент массива с индексом =i,
```

```
    // после чего увеличиваем i на единицу
```

```
for (i=0; i<=7; i++)
```

```
{
```

```
    printf ("%d, ", mas[i] );
```

```
}
```

```
printf ("%d\n", mas[8] );
```

```
    //вводим первую матрицу 2x2 и заполняем ее значениями {1, 1, 1, 4}
```

```
int mat1[2][2];
```

```
mat1[0][0]= 1;
```

```
mat1[0][1]= 1;
```

```
mat1[1][0]= 1;
```

```
mat1[1][1]= 4;
```

```
    //вводим вторую матрицу 2x2 со значениями {1, 2, 4, 1}
```

```
int mat2[2][2];
```

```
mat2[0][0]= 1;
```

```
mat2[0][1]= 2;
```

```
mat2[1][0]= 4;
```

```
mat2[1][1]= 1;
```

```
    //вводим третью, пустую матрицу 2x2 куда впишем результаты сложения
```

```
int mat3[2][2];
```

```
    //.. присваиваем элементам третьей матрицы результаты арифметических операций
```

```
mat3[0][0] = mat1[0][0]*mat2[0][0] + mat1[0][1]*mat2[1][0];
```

```
mat3[0][1] = mat1[0][0]*mat2[0][1] + mat1[0][1]*mat2[1][1];
```

```
mat3[1][0] = mat1[1][0]*mat2[0][0] + mat1[1][1]*mat2[1][0];
```

```
mat3[1][1] = mat1[1][0]*mat2[0][1] + mat1[1][1]*mat2[1][1];
```

```
    // "пока значения i и j не больше 1 мы выписываем в строчку значения mat3[i][j],
```

```
    // а после переходим на вторую строчку, тем самым образуем матрицу"
```

```
for (i=0; i<=1; i++)
```

```
{
```

```
    for (j=0; j<=1; j++)
```

```
{
```

```
    printf ("%d ", mat3[i][j]);
```

```
}
```

```
printf("\n");
```

```
}
```

```
return 0;
```

```
}
```

Задание 6

```
#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>

int main()
{
    char arr[4], *dyn_arr;
    int i;
    // заполняем массив буквами
    arr[0] = 'H';
    arr[1] = 'e';
    arr[2] = 'l';
    arr[3] = 'l';

    //выводим получившееся слово
    for ("%c", i=0; i<4; i++)
    {
        printf("%c", arr[ i ]);
    }
    printf ("\n");

    //выделяем память (malloc) под 4 элемента массива типа char
    dyn_arr =(char*) malloc (4 * sizeof(char));
    dyn_arr [0] = 'H';
    dyn_arr [1] = 'e';
    dyn_arr [2] = 'l';
    dyn_arr [3] = 'l';

    //печатаем то же слово, но с помощью динамической памяти
    for (i = 0; i<4; i++)
    {
        printf("%c ", dyn_arr[i] );
    }
    //очищаем блок памяти, выделенный ранее функцией "malloc"
    free(dyn_arr);
    return 0;
}
```

Задание 7

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
enum enter_forms
{
    passenger,
    truck,
    bus,
    racing
};
struct square_square
{
    int x1,x2,y1,y2;
};
union
{
    struct
    {
        unsigned ready: 1;
        unsigned little_toner : 1;
        unsigned damaged_drum: 1;
        unsigned no_paper : 1;
    } bytes;
    unsigned a ;
}
task3;
int main()
{
    enum enter_forms task1 ;
    task1 = bus;
    printf ("%d\n\n", task1 );
    struct square_square S;
    float task2, x1,x2,y1,y2,a,b ;
    x1 = 1; y1 = 3; x2 = -2; y2 = 5;
    a= pow(x1-x2, 2);
    b= pow(y1-y2, 2);
    task2= sqrt(a + b);
    printf ("%f\n\n" , task2 );
    scanf ("%x" , &task3.a );
    printf ("%d %d %d %d", task3.bytes.ready, task3.bytes.little_toner, task3.bytes.damaged_drum,
    task3.bytes.no_paper);
    return 0;
}
```

Задание 8

```
#include <stdio.h>
#include <string.h>
int main ()
{
    int len_str, i;
    char *k;
    char *isym;
    char str1[255];
    char str2[255];

    scanf("%s\n%s\n%s", str1, str2, i);
    // наглое сложение строк
    printf("%s%s\n", str1, str2);

    //сравн
    if (strncmp (str1, str2, i)==0)
        printf("%d\n", 1);
    else
        printf("%d\n", 0);

    //длина строки
    len_str = strlen(str1);
    printf("%d\n", len_str);

    //поиск в одной строке любого из множества символов, входящих во вторую строку
    isym = strpbrk(str1, str2);
    if (isym == NULL)
        printf("%d\n", 0); //общих нет
    else
        printf("%d\n", 1); //существуют общие

    //длина отрезка одной строки, не содержащая символы второй строки
    printf("%d\n", strcspn (str2,str1));
    return 0;
}
```

*Лабораторные работы 5-8
выполнил студент 1 курса ИС
Микаилов Михаил Аскерович
3110/14 вариант*