



**Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Московский государственный технический
университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и
управления»**

Лабораторная работа №6 по курсу
«Базовые компоненты интернет технологий»

Выполнила:
студент группы № ИУ5-33Б
Балюк А.В

Проверил:
Преподаватель
Гапанюк Ю.Е

2022 г.

Задание:

1. Реализуйте любое из заданий курса на языке программирования Rust.
2. Разработайте хотя бы один макрос.
3. Разработайте модульные тесты (не менее 3 тестов).

Листинг:

```
use std::io;

#[cfg(test)]
mod tests{
    use super::main;
    use super::*;
    #[test]
    fn test1(){
        let mut a:f64 = 1.0;
        let mut b:f64 = -5.0;
        let mut c:f64 = -36.0;
        let mut d:f64 = b.powf(2.0) - 4.0 *(a*c);
        let mut need:Vec<f64>=Vec::new();
        need.push(3.0);
        need.push(-3.0);
        assert_eq!(task1(a,b,c),need);
    }
    #[test]
    fn test2(){
        let mut a:f64 = 1.0;
        let mut b:f64 = -5.0;
        let mut c:f64 = 4.0;
        let mut d:f64 = b.powf(2.0) - 4.0 *(a*c);
        let mut need:Vec<f64>=Vec::new();
        need.push(2.0);
        need.push(-2.0);
        need.push(1.0);
        need.push(-1.0);
        assert_eq!(task1(a,b,c),need);
    }
    #[test]
    fn test3(){
        let mut a:f64 = -4.0;
        let mut b:f64 = 16.0;
        let mut c:f64 = 0.0;
        let mut d:f64 = b.powf(2.0) - 4.0 *(a*c);
        let mut need:Vec<f64>=Vec::new();
        need.push(0.0);
        need.push(2.0);
        need.push(-2.0);
        assert_eq!(task1(a,b,c),need);
    }
}
```

```

    }
}

fn task1(a:f64, b:f64, c:f64)->Vec<f64>{
    let d:f64 = b.powf(2.0) - 4.0 *(a*c);
    let mut R_ans:Vec<f64>=Vec::new();
    if (d>0.0){
        let x1 = (-b+d.sqrt())/(2.0*a);
        let x2 = (-b-d.sqrt())/(2.0*a);
        if (x1<0.0) {
            if (x2<0.0)
            {
                R_ans.push(-1.0);
            }
            else {
                if (x2==0.0){
                    R_ans.push(0.0);
                }
                else {
                    R_ans.push(x2.sqrt());
                    R_ans.push(-x2.sqrt());
                }
            }
        }
        else if (x2<0.0){
            if (x1<0.0)
            {
                R_ans.push(-1.0);
            }
            else {
                if (x1==0.0){
                    R_ans.push(0.0);
                }
                else {
                    R_ans.push(x1.sqrt());
                    R_ans.push(-x1.sqrt());
                }
            }
        }
    }
    else
    {
        if (x1==0.0){
            let ans1 = 0.0;
            let ans2 = x2.sqrt();
            let ans3= -x2.sqrt();
            R_ans.push(0.0);
            R_ans.push(x2.sqrt());

```

```

        R_ans.push(-x2.sqrt());

    }
    else if (x2==0.0)
    {
        let ans1 = x1.sqrt();
        let ans2 = -x1.sqrt();
        let ans3= 0.0;
        R_ans.push(0.0);
        R_ans.push(x1.sqrt());
        R_ans.push(-x1.sqrt());

    }
    else{
        let ans1 = x1.sqrt();
        let ans2 = -x1.sqrt();
        let ans3= x2.sqrt();
        let ans4 = -x2.sqrt();
        R_ans.push(x1.sqrt());
        R_ans.push(-x1.sqrt());
        R_ans.push(x2.sqrt());
        R_ans.push(-x2.sqrt());
    }
    return R_ans;
}
}
else
if (d==0.0){
    if (b==0.0 || b<0.0)
    {
        R_ans.push(0.0);
    }
    else{
        let x = -b/(2.0*a);
        let Sans1 = x.sqrt();
        let Sans2 = -x.sqrt();
        R_ans.push(x.sqrt());
        R_ans.push(-x.sqrt());
    }
}
if (d<0.0){
    R_ans.push(-1.0);
}
return R_ans;
}

macro_rules! print_answers {
    ($d:expr, $b:expr, $a:expr) => {
        if ($d>0.0){
            let x1 = (-$b+$d.sqrt()/(2.0*$a);
            let x2 = (-$b-$d.sqrt()/(2.0*$a);

```

```

    if (x1<0.0) {
        if (x2==0.0){
            println!("Корень уравнения: 0");
        }
        else {
            println!("Корень уравнения: {} и {}",x2.sqrt(),-x2.sqrt())
        }
    }
    else if (x2<0.0){
        if (x1==0.0){
            println!("Корень уравнения: 0");
        }
        else {
            println!("Корень уравнения: {} и {}",x1.sqrt(),-x1.sqrt())
        }
    }
    else
    {
        if (x1==0.0){
            let ans1 = 0.0;
            let ans2 = x2.sqrt();
            let ans3= -x2.sqrt();

            println!("Корни уравнения: {}, {}, {}",ans1,ans2, ans3);
        }
        else if (x2==0.0)
        {
            let ans1 = x1.sqrt();
            let ans2 = -x1.sqrt();
            let ans3= 0;
            println!("Корни уравнения: {}, {}, {}",ans1,ans2, ans3);
        }
        else{
            let ans1 = x1.sqrt();
            let ans2 = -x1.sqrt();
            let ans3= x2.sqrt();
            let ans4 = -x2.sqrt();
            println!("Корни уравнения: {}, {}, {} и {}",ans1,ans2, ans3, ans4);
        }
    }
}
}
else
if ($d==0.0){
    if ($b==0.0 || $b<0.0)
    {
        println!("Корень уравнения: 0");
    }
    else{
        let x = -$b/(2.0*$a);
        let Sans1 = x.sqrt();
        let Sans2 = -x.sqrt();

```

```

        println!("Корни уравнения: {} и {}", Sans1, Sans2)
    }
}
if ($d<0.0){
    println!("Корней нет");
}
};
}

fn main() {

    let mut A = String::new();
    let mut B = String::new();
    let mut C = String::new();

    let mut a:f64=0.0;
    let mut b:f64=0.0;
    let mut c:f64=0.0;

    println!("Введите корни биквадратного уравнения");

    loop{
        println!("Коэффициент A: ");
        A.clear();
        match io::stdin().read_line(&mut A) {
            Ok(_)=>{},
            Err(e)=>{println!("Ошибка ввода параметра A - {}. Потворите попытку",e)}
        }
        a = match A.trim().parse() {
            Ok(num)=>num,
            Err(_)=>continue,
        };
        break;
    }

    loop {
        println!("Коэффициент B: ");
        B.clear();
        match io::stdin().read_line(&mut B) {
            Ok(_)=>{},
            Err(e)=>{println!("Ошибка ввода параметра B - {}. Потворите попытку",e)}
        }
        b = match B.trim().parse() {
            Ok(num)=>num,
            Err(_)=>continue,
        };
        break;
    }

    loop {

```

```

println!("Коэффициент C: ");
C.clear();
match io::stdin().read_line(&mut C) {
    Ok(_)=>{},
    Err(e)=>{println!("Ошибка ввода параметра C - {}. Потворите попытку",e)}
}
C = match C.trim().parse() {
    Ok(num)=>num,
    Err(_)=>continue,
};
break;
}

let mut answer:Vec<f64> = task1(a,b,c);
if (answer[0]==-1.0)|| (answer.len()==0){
    println!("Корней нет");
}
else {
    print!("Корни уравнения: ");
    for i in &answer{
        print!("{}", i);
    }
}

//let d:f64 = b.powf(2.0) - 4.0 *(a*c);
//print_answers!(d,b,a);
}

```

Результат:

```

Введите корни биквадратного уравнения
Коэффициент A:
1
Коэффициент B:
-5
Коэффициент C:
3
Корни уравнения: 2.074313293051943 -2.074313293051943 0.8349996181244669 -0.8349996181244669 * T
|

running 3 tests
test tests::test2 ... ok
test tests::test1 ... ok
test tests::test3 ... ok

test result: ok. 3 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

```