МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации

**

**ОТЧЁТ**

**по лабораторной работе №2**

**«**Классификация языков программирования»

**по дисциплине: «***Программирование***»**

Выполнил:Проверил:

Студент гр. «АБс-324», «АВТФ» *ассистент кафедры ЗИ*

*Зенцов А.С. Исаев Г. А.*

«\_\_» \_\_\_\_\_\_ 2024 г.«\_\_\_» \_\_\_\_\_\_ 2024 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (подпись)

Новосибирск 2024

**Цели и задачи работы**: изучение функций ввода-вывода данных, программирования вычисления значения выражения.

**Задание к работе**:

Реализовать линейный вычислительный процесс. Самостоятельно решить задачу в соответствии с индивидуальным вариантом. Реализовать представленные задачи на языках программирования (перечень ЯП в файле)

Реализовать линейный вычислительный процесс любого задания на языке программирования Assembler, Haskell.

Представленные задачи можно реализовать на каждом языке в одной программе с последовательным выполнением.

Методика выполнения работы:

1. Определить типы используемых в программе данных.
2. Описать переменные.
3. Написать функции ввода-вывода.
4. Разработать алгоритм решения задачи по индивидуальному заданию.
5. Написать и отладить программу с вводом-выводом информации
6. Протестировать работу программы на различных исходных данных.
7. Изменить формат вывода, проверить работу программы при другом формате вывода.

**Методика выполнения работы**:

1. Разработать алгоритм решения задачи по индивидуальному заданию.
2. Написать и отладить программу решения задачи.
3. Протестировать работу программы на различных исходных данных.
4. Продемонстрировать работу с Git и GitHub на примере написанной программы.
5. По запросу преподавателя быть готовым модифицировать/добавить функционал программы.
6. Ответить на теоретические вопросы к лабораторной работе на выбор преподавателя.

**Ход работы**

Задание 1. Вариант 10

10. Существует шахматная доска NxM, по которой ходит фигура. Изначально фигура стоит на одной из клеток доски. Задана строка s с командами, задающими направление ходов (L, R, D, U). Один ход = перемещение на 1 клетку. Необходимо определить, возможно ли совершить все заданные команды, не выходя за пределы шахматной доски. Если это возможно, вывести клетку, с которой фигура начинает движение. Пример: N = 1, M = 1, s = "L". Результат: No. (шахматная доска состоит из одной клетки, при перемещении влево фигура выйдет за ее пределы) N = 3, M = 2, s = "DDR". Результат: (1,1).

**Описание программы**

Представим шахматную доску в виде матрицы с n строк и m столбцов. На вход получаем n, m и последовательность ходов (переменная hodi). Во внешнем цикле идем по строкам, во внутреннем – по столбцам. В функцию moves поступает пара значений – позиция фигуры на доске, далее с помощью конструкции switch – case (if – else на некоторых ЯП) двигаем фигуру относительно ячеек матрицы. Возвращаем значение – конечное положение фигуры. Если после исполнения команд фигура не окажется за пределами – программа выводит значение клетки, из которой может пойти фигура. В противном случае, запускается следующая итерация. Если нет такой позиции, то программа выводит «No».

Если в строку поступает неизвестная команда, то программа выводит «<hod> is undefined» и завершает ее исполнение.

**Листинг программы на C++**

#include <iostream>

#include <string>

using namespace std;

// функция, которая возвращает конечную позицию фигуры

pair<int, int> moves(string hodi, int n, int m) {

// перебор каждого действия из строки

for (char hod : hodi) {

switch (hod) {

case 'L':

m--;

break;

case 'R':

m++;

break;

case 'D':

n++;

break;

case 'U':

n--;

break;

default:

cout << hod << " is undefined" << endl;

pair<int, int> p = make\_pair(-123, 0);

return p;

break;

}

}

pair<int, int> p = make\_pair(n,m);

return p;

}

int main() {

int n, m = 0;

cout << "n = ";

cin >> n;

while (cin.fail() || n < 0) {

system("cls");

cout << "Enter INTEGER > 0" << endl;

cin.clear();

cin.ignore();

cin >> n;

}

cout << "m = ";

cin >> m;

while (cin.fail() || m < 0) {

system("cls");

cout << "Enter INTEGER > 0" << endl;

cin.clear();

cin.ignore();

cin >> m;

}

cout << "moves = ";

string hodi;

cin >> hodi;

// идем по строкам

for (int i = 1; i <= n; i++) {

// идем по столбцам

for (int j = 1; j <= m; j++) {

pair<int,int> p = moves(hodi, i, j);

// если фигура не вышла за пределы доски - переход к метке exist

if ((p.first > 0 && p.first <= n) && (p.second > 0 && p.second <= m)){

cout << i << " " << j << endl;

goto exist;

}

// если неправильная команда - переход к метке doenstexist

if (p.first == -123) {

goto doesntexist;

}

}

}

// метка doesntexist, выводит No

doesntexist:

cout << "No" << endl;

// метка label, завершает выполнение программы

exist:

return 0;

}

**Листинг программы на Python**

def moves(hodi, n, m):  
 for hod in hodi:  
 if hod == 'L':  
 m -= 1  
 if hod == 'R':  
 m += 1  
 if hod == 'D':  
 n += 1  
 if hod == 'U':  
 n -= 1  
 else:  
 print(hod, " is undefined")  
 return -123, 0  
 return n, m  
  
  
print("n = ")  
n = int(input())  
print("m = ")  
m = int(input())  
print("moves = ")  
hodi = str(input())  
isDone = False  
for i in range(1, n + 1):  
 for j in range(1, m + 1):  
 nnew, mnew = moves(hodi, i, j)  
 if ((nnew > 0 and nnew <= n) and (mnew > 0 and mnew <= m)):  
 print(i, j)  
 isDone = True  
 break  
 if nnew == -123:  
 print('No')  
 exit()  
 if isDone:  
 break  
if not isDone:  
 print('No')

**Листинг программы на Go**

package main

import (

"fmt"

)

func moves(hodi string, n, m int) (int, int) {

for \_, hod := range hodi {

switch hod {

case 'L':

m--

case 'R':

m++

case 'D':

n++

case 'U':

n--

default:

fmt.Printf("%c is undefined\n", hod)

return -123, 0

}

}

return n, m

}

func main() {

var n, m int

fmt.Print("n = ")

fmt.Scan(&n)

fmt.Print("m = ")

fmt.Scan(&m)

var hodi string

fmt.Scan(&hodi)

for i := 1; i <= n; i++ {

for j := 1; j <= m; j++ {

nnew, mnew := moves(hodi, i, j)

if n > 0 && nnew <= n && m > 0 && mnew <= m {

fmt.Printf("%d %d\n", i, j)

goto exist

}

if nnew == -123 {

goto doesntexist

}

}

}

doesntexist:

fmt.Println("No")

exist:

return

}

**Листинг программы на JavaScript**

function moves(hodi, n, m) {

for (let hod of hodi) {

switch (hod) {

case 'L':

m--;

break;

case 'R':

m++;

break;

case 'D':

n++;

break;

case 'U':

n--;

break;

default:

console.log(hod + " is undefined");

return [-123, 0];

}

}

return [n, m];

}

function findFigurePosition(n, m, hodi) {

for (let i = 1; i <= n; i++) {

for (let j = 1; j <= m; j++) {

let [x, y] = moves(hodi, i, j);

if (x > 0 && x <= n && y > 0 && y <= m) {

return [i, j];

}

if (x === -123) {

return "No";

}

}

}

return "No";

}

let n = parseInt(prompt("n = "));

let m = parseInt(prompt("m = "));

let hodi = prompt("Enter the moves: ");

let result = findFigurePosition(n, m, hodi);

console.log(result);

**Листинг программы на Swift**

import Foundation

func moves(hodi: String, n: Int, m: Int) -> (Int, Int) {

var n = n

var m = m

for hod in hodi {

switch hod {

case "L":

m -= 1

case "R":

m += 1

case "D":

n += 1

case "U":

n -= 1

default:

print("\(hod) is undefined")

return (-123, 0)

}

}

return (n, m)

}

func main() {

var n = 0

var m = 0

print("n = ")

if let input = readLine(), let inputInt = Int(input) {

n = inputInt

}

print("m = ")

if let input = readLine(), let inputInt = Int(input) {

m = inputInt

}

var hodi = ""

print("moves: ")

if let input = readLine() {

hodi = input

}

for i in 1...n {

for j in 1...m {

let p = moves(hodi: hodi, n: i, m: j)

if (p.0 > 0 && p.0 <= n) && (p.1 > 0 && p.1 <= m) {

print("\(i) \(j)")

return

}

if p.0 == -123 {

print("No")

return

}

}

}

print("No")

}

main()

**Листинг программы на C#**

using System;

namespace ChessMoves

{

class Program

{

static Tuple<int, int> Moves(string hodi, int n, int m)

{

foreach (char hod in hodi)

{

switch (hod)

{

case 'L':

m--;

break;

case 'R':

m++;

break;

case 'D':

n++;

break;

case 'U':

n--;

break;

default:

Console.WriteLine(hod + " is undefined");

return Tuple.Create(-123, 0);

}

}

return Tuple.Create(n, m);

}

static void Main(string[] args)

{

int n, m = 0;

Console.Write("n = ");

n = int.Parse(Console.ReadLine());

Console.Write("m = ");

m = int.Parse(Console.ReadLine());

string hodi = Console.ReadLine();

for (int i = 1; i <= n; i++)

{

for (int j = 1; j <= m; j++)

{

Tuple<int, int> p = Moves(hodi, i, j);

if (p.Item1 > 0 && p.Item1 <= n && p.Item2 > 0 && p.Item2 <= m)

{

Console.WriteLine(i + " " + j);

goto exist;

}

if (p.Item1 == -123)

{

goto doesntexist;

}

}

}

doesntexist:

Console.WriteLine("No");

exist:

return;

}

}

}

**Листинг программы на Java**

import java.util.Scanner;

public class ChessMoves {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("n = ");

int n = scanner.nextInt();

System.out.print("m = ");

int m = scanner.nextInt();

System.out.print("Moves: ");

String moves = scanner.next();

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= m; j++) {

if (FuncMoves(moves, i, j, n, m)) {

System.out.println(i + " " + j);

return;

}

}

}

System.out.println("No");

}

public static boolean FuncMoves(String moves, int n, int m, int boardSizeN, int boardSizeM) {

for (char move : moves.toCharArray()) {

switch (move) {

case 'L':

m--;

break;

case 'R':

m++;

break;

case 'D':

n++;

break;

case 'U':

n--;

break;

default:

System.out.println(move + " is undefined");

return false;

}

}

return n > 0 && n <= boardSizeN && m > 0 && m <= boardSizeM;

}

}

**Листинг программы на Ruby**

def moves(hodi, n, m)

hodi.each\_char do |hod|

case hod

when 'L'

m -= 1

when 'R'

m += 1

when 'D'

n += 1

when 'U'

n -= 1

else

puts "#{hod} is undefined"

return [-123, 0]

end

end

[n, m]

end

n = 0

m = 0

puts "n = "

n = gets.chomp.to\_i

puts "m = "

m = gets.chomp.to\_i

puts "moves = "

hodi = gets.chomp

(1..n).each do |i|

(1..m).each do |j|

p = moves(hodi, i, j)

if p[0] > 0 && p[0] <= n && p[1] > 0 && p[1] <= m

puts "#{i} #{j}"

abort

end

if p[0] == -123

abort "No"

end

end

end

puts "No"

**Листинг программы на Rust**

use std::io;

fn moves(hodi: &str, n: i32, m: i32) -> (i32, i32) {

let mut n = n;

let mut m = m;

for hod in hodi.chars() {

match hod {

'L' => m -= 1,

'R' => m += 1,

'D' => n += 1,

'U' => n -= 1,

\_ => {

println!("{} is undefined", hod);

return (-123, 0);

}

}

}

(n, m)

}

fn main() {

let mut n = String::new();

let mut m = String::new();

println!("n = ");

io::stdin().read\_line(&mut n).expect("Failed to read input");

println!("m = ");

io::stdin().read\_line(&mut m).expect("Failed to read input");

let n: i32 = n.trim().parse().expect("Invalid input");

let m: i32 = m.trim().parse().expect("Invalid input");

let mut hodi = String::new();

io::stdin().read\_line(&mut hodi).expect("Failed to read input");

let hodi = hodi.trim();

for i in 1..=n {

for j in 1..=m {

let p = moves(hodi, i, j);

if (p.0 > 0 && p.0 <= n) && (p.1 > 0 && p.1 <= m) {

println!("{} {}", i, j);

std::process::exit(0)

}

if p.0 == -123 {

println!("No");

std::process::exit(0)

}

}

}

println!("No");

}

**Листинг программы на Kotlin**

import java.util.Scanner

fun moves(hodi: String, n: Int, m: Int): Pair<Int, Int> {

var nloc = n

var mloc = m

for (hod in hodi) {

when (hod) {

'L' -> mloc--

'R' -> mloc++

'D' -> nloc++

'U' -> nloc--

else -> {

println("$hod is undefined")

return Pair(-123, 0)

}

}

}

return Pair(nloc, mloc)

}

fun main() {

val scanner = Scanner(System.`in`)

println("n = ")

var n = 0

n = scanner.nextInt()

println("m = ")

var m = 0

m = scanner.nextInt()

println("moves: ")

var hodi = ""

hodi = scanner.nextLine()

for (i in 1..n) {

for (j in 1..m) {

val (nloc, mloc) = moves(hodi, i, j)

if (nloc in 1..n && mloc in 1..m) {

println("$i $j")

return

}

if (nloc == -123) {

println("No")

return

}

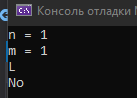
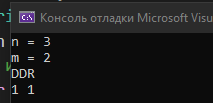
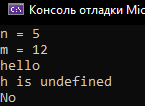
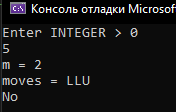
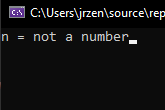
}

}

println("No")

}

**Результат работы программы**

**   
**

Задание 2. Вариант 10

У Вас есть n банок с водой. Обмену на одну полную банку подлежат m пустых банок. Необходимо вычислить максимальное число банок, которое пользователь может выпить до положения количества пустых банок, равного одной. А также количество итераций. Отобразить пошаговый процесс (как в примере 1, символы можно изменить в программе).

**Листинг программы на C++**

#include <iostream>

using namespace std;

void main() {

// есть полных банок

int n = 0;

cout << "full bottles: ";

cin >> n;

// проверка на дурака

while (cin.fail() || n < 0) {

system("cls");

cout << "Enter INTEGER > 0" << endl;

cin.clear();

cin.ignore();

cin >> n;

}

// пустых банок на обмен

int m = 0;

cout << "empty bottles for exchange: ";

cin >> m;

// проверка на дурака

while (cin.fail() || n < 0) {

system("cls");

cout << "Enter INTEGER > 0" << endl;

cin.clear();

cin.ignore();

cin >> n;

}

// пустые банки после итерации

int mc = n;

// переменная для проверки возможности обменяться

int mcount = n;

// счетчик итераций (выпил + обменял)

int count = 0;

// оставшиеся пустые банки после обмена

int ost = 0;

// счетчик выпитых банок

int drinkedBottles = 0;

while (mcount > 0) {

cout << string(n, 'F') << endl;

drinkedBottles += n;

// выпил

count++;

mc = n + ost;

cout << string(mc, 'E') << endl;

ost = mc % m;

n = mc / m;

// обменял

count++;

mc = ost;

mcount = n;

}

cout << drinkedBottles << " " << count-1 << endl;

}

**Листинг программы на Python**

n = int(input("full bottles: "))  
m = int(input("empty bottles for exchange: "))  
mc = n  
mcount = n  
count = 0  
ost = 0  
drinkedBottles = 0  
  
while mcount > 0:  
 print('F' \* n)  
 drinkedBottles += n  
 count += 1  
 mc = n + ost

print('E' \* mc)  
 ost = mc % m  
 n = mc // m  
 count += 1  
 mc = ost  
 mcount = n  
  
print(drinkedBottles, count-1)

**Листинг программы на Go**

package main

import "fmt"

func main() {

var n, m int

fmt.Print("full bottles: ")

fmt.Scan(&n)

fmt.Print("empty bottles for exchange: ")

fmt.Scan(&m)

mc := n

mcount := n

count := 0

ost := 0

drinkedBottles := 0

for mcount > 0 {

for i := 0; i < n; i++{

fmt.Printf("F")

}

fmt.Print("\n")

drinkedBottles += n

count++

mc = n + ost

for i := 0; i < mc; i++{

fmt.Printf("E")

}

fmt.Print("\n")

ost = mc % m

n = mc / m

count++

mc = ost

mcount = n

}

fmt.Println(drinkedBottles, count-1)

}

**Листинг программы на JavaScript**

function main() {

let n = 0;

console.log("full bottles: ");

n = parseInt(prompt());

let m = 0;

console.log("empty bottles for exchange: ");

m = parseInt(prompt());

let mc = n;

let mcount = n;

let count = 0;

let ost = 0;

let drinkedBottles = 0;

while (mcount > 0) {

console.log("F".repeat(n));

drinkedBottles += n;

count++;

mc = n + ost;

console.log("E".repeat(mc));

ost = mc % m;

n = Math.floor(mc / m);

count++;

mc = ost;

mcount = n;

}

console.log(drinkedBottles + " " + (count - 1));

}

main();

**Листинг программы на Swift**

import Foundation

func main() {

var n = 0

print("full bottles: ")

if let input = readLine() {

if let intValue = Int(input) {

n = intValue

}

}

var m = 0

print("empty bottles for exchange: ")

if let input = readLine() {

if let intValue = Int(input) {

m = intValue

}

}

var mc = n

var mcount = n

var count = 0

var ost = 0

var drinkedBottles = 0

while mcount > 0 {

print(String(repeating: "F", count: n))

drinkedBottles += n

count += 1

mc = n + ost

print(String(repeating: "E", count: mc))

ost = mc % m

n = mc / m

count += 1

mc = ost

mcount = n

}

print(drinkedBottles, count-1)

}

main()

**Листинг программы на C#**

using System;

class Program

{

static void Main()

{

int n = 0;

Console.Write("full bottles: ");

n = int.Parse(Console.ReadLine());

int m = 0;

Console.Write("empty bottles for exchange: ");

m = int.Parse(Console.ReadLine());

int mc = n;

int mcount = n;

int count = 0;

int ost = 0;

int drinkedBottles = 0;

while (mcount > 0)

{

Console.WriteLine(new string('F', n));

drinkedBottles += n;

count++;

mc = n + ost;

Console.WriteLine(new string('E', mc));

ost = mc % m;

n = mc / m;

count++;

mc = ost;

mcount = n;

}

Console.WriteLine(drinkedBottles + " " + (count - 1));

}

}

**Листинг программы на Java**

import java.util.Scanner;

public class BottleExchange {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("full bottles: ");

int n = scanner.nextInt();

System.out.print("empty bottles for exchange: ");

int m = scanner.nextInt();

int mc = n;

int mcount = n;

int count = 0;

int ost = 0;

int drinkedBottles = 0;

while (mcount > 0) {

System.out.println("F".repeat(n));

drinkedBottles += n;

count++;

mc = n + ost;

System.out.println("E".repeat(mc));

ost = mc % m;

n = mc / m;

count++;

mc = ost;

mcount = n;

}

System.out.println(drinkedBottles + " " + (count - 1));

}

}

**Листинг программы на Ruby**

puts "full bottles: "

n = gets.chomp.to\_i

puts "empty bottles for exchange: "

m = gets.chomp.to\_i

mc = n

mcount = n

count = 0

ost = 0

drinkedBottles = 0

while mcount > 0

puts "F" \* n

drinkedBottles += n

count += 1

mc = n + ost

puts "E" \* mc

ost = mc % m

n = mc / m

count += 1

mc = ost

mcount = n

end

puts "#{drinkedBottles} #{count-1}"

**Листинг программы на Rust**

use std::io;

fn main() {

let mut n: i32 = {

let mut input = String::new();

println!("full bottles: ");

io::stdin().read\_line(&mut input).expect("Type only 1 NUMBER!");

input.trim().parse().expect("Type only 1 NUMBER!")

};

let mut m: i32 = {

let mut input = String::new();

println!("empty bottles for exchange: ");

io::stdin().read\_line(&mut input).expect("Type only 1 NUMBER!");

input.trim().parse().expect("Type only 1 NUMBER!")

};

let mut mc = n;

let mut mcount = n;

let mut count = 0;

let mut ost = 0;

let mut drinkedBottles = 0;

while mcount > 0 {

println!("{}", "F".repeat(n as usize));

drinkedBottles += n;

count += 1;

mc = n + ost;

println!("{}", "E".repeat(mc as usize));

ost = mc % m;

n = mc / m;

count += 1;

mc = ost;

mcount = n;

}

println!("{} {}", drinkedBottles, count - 1);

}

**Листинг программы на Kotlin**

import java.util.Scanner

fun main() {

val scanner = Scanner(System.`in`)

printLn("full bottles: ")

var n = 0

n = scanner.nextInt()

printLn("empty bottles for exchange: ")

var m = 0

m = scanner.nextInt()

var mc = n

var mcount = n

var count = 0

var ost = 0

var drinkedBottles = 0

while (mcount > 0) {

println("F".repeat(n))

drinkedBottles += n

count++

mc = n + ost

println("E".repeat(mc))

ost = mc % m

n = mc / m

count++

mc = ost

mcount = n

}

println("$drinkedBottles ${count-1}")

}

**Листинг программы на Assembler**

global \_start

section .data

n dq 35

m dq 7

section .text

\_start:

push rbp

mov rbp, rsp

mov rbx, [n]

mov rax, [n]

mov rcx, 0

whil:

mov [rbp-8], ah

add rax, [rbp-8]

mov ah, 0

div byte [m]

add rbx, rax

add rcx, 2

cmp al, 0

jg whil

sub rcx, 1

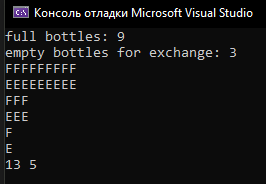
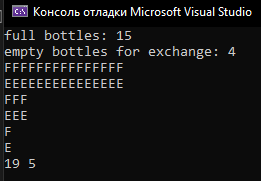
;для вывода count поменять регистр rbx на rcx

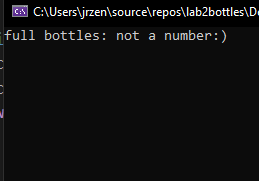
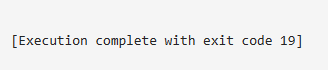
mov rdi, rbx

mov rax, 60

syscall

**Результат работы программы**

Задание 3. Вариант 10

10. Для каждого числа из последовательности определить, можно ли поставить внутри числа символ «:» таким образом, чтобы получилась корректная запись времени в формате часы:минуты. Пример. Вход: 3 1244 322 999 Выход: YES YES NO.

**Описание программы**

Программа состоит из трех функций. WhatTimeIsIt определяет количество часов и минут в заданном числе и возвращает True или False в зависимости, удовлетворяет ли число формату времени Hours : Minutes. YesOrNo выводит YES или NO, если из предыдущей функции вернулось значение True или False, соответственно. В функции main происходит ввод количества чисел, самих чисел.

**Листинг программы на C++**

#include <iostream>

#include <vector>

using namespace std;

// функция определяет, является ли число временем в формате Hours:Minutes

bool WhatTimeIsIt(int number) {

int hours = number / 100;

int minutes = number % 100;

return (hours >= 0 && hours <= 23) && (minutes >= 0 && minutes <= 59);

}

// получает значение из функции WhatTimeIsIt, если true -> YES, если false -> NO

void YesOrNo(vector<int> numbers, int n) {

for (int i = 0; i < n; i++) {

if (WhatTimeIsIt(numbers[i])) {

cout << "YES ";

}

else {

cout << "NO ";

}

}

cout << endl;

}

int main() {

// количество чисел

int n = 0;

cout << "n = ";

cin >> n;

while (cin.fail() || n < 0) {

system("cls");

cout << "Enter INTEGER > 0" << endl;

cin.clear();

cin.ignore();

cin >> n;

}

// вектор с числами

vector<int> numbers;

int x = 0;

// ввод n чисел

for (int i = 0; i < n; i++) {

cout << "Enter " << i + 1 << " number: ";

cin >> x;

while (cin.fail() || x < 0) {

system("cls");

cout << "Enter INTEGER > 0" << endl;

cin.clear();

cin.ignore();

cin >> x;

}

numbers.push\_back(x);

}

YesOrNo(numbers, n);

}

**Листинг программы на Python**

def WhatTimeIsIt(number):  
 hours = number / 100  
 minutes = number % 100  
 return (hours >= 0 and hours <= 23) and (minutes >= 0 and minutes <= 59)  
  
def YesOrNo(numbers, n):  
 for i in range(n):  
 if WhatTimeIsIt(numbers[i]):  
 print("YES")  
 else:  
 print("NO")  
  
n = int(input("n = "))  
numbers = list()  
for i in range(n):  
 x = int(input("Enter the number: "))  
 numbers.append(x)  
YesOrNo(numbers, n)

**Листинг программы на Go**

package main

import (

"fmt"

)

func WhatTimeIsIt(number int) bool {

hours := number / 100

minutes := number % 100

return (hours >= 0 && hours <= 23) && (minutes >= 0 && minutes <= 59)

}

func YesOrNo(number int, n int) {

if WhatTimeIsIt(number) {

fmt.Println("YES")

} else {

fmt.Println("NO")

}

}

func main() {

var n int

fmt.Print("n = ")

fmt.Scan(&n)

for i := 0; i < n; i++ {

var x int

fmt.Print("Enter the number: ")

fmt.Scan(&x)

YesOrNo(x, n)

}

}

**Листинг программы на JavaScript**

function WhatTimeIsIt(number){

let hours = Math.floor(number / 100);

let minutes = number % 100;

return (hours >= 0 && hours <= 23) && (minutes >= 0 && minutes <= 59);

}

function YesOrNo(number, n){

if (WhatTimeIsIt(number)) {

console.log("YES");

} else {

console.log("NO");

}

}

let n = parseInt(prompt("n = "));

for (let i = 0; i < n; i++) {

let x = parseInt(prompt("Enter the number: "));

YesOrNo(x, n);

}

**Листинг программы на Swift**

func whatTimeIsIt(number: Int) -> Bool {

let hours = number / 100

let minutes = number % 100

return (hours >= 0 && hours <= 23) && (minutes >= 0 && minutes <= 59)

}

func yesOrNo(number: Int, n: Int) {

if whatTimeIsIt(number: number) {

print("YES")

} else {

print("NO")

}

}

print("n = ", terminator: "")

if let n = Int(readLine() ?? "") {

for \_ in 0..<n {

print("Enter the number: ", terminator: "")

if let x = Int(readLine() ?? "") {

yesOrNo(number: x, n: n)

}

}

}

**Листинг программы на C#**

using System;

public class Program{

public static bool WhatTimeIsIt(int number){

int hours = number / 100;

int minutes = number % 100;

return (hours >= 0 && hours <= 23) && (minutes >= 0 && minutes <= 59);

}

public static void YesOrNo(int number, int n){

if (WhatTimeIsIt(number)){

Console.WriteLine("YES");

}

else

{

Console.WriteLine("NO");

}

}

public static void Main(){

Console.Write("n = ");

int n = int.Parse(Console.ReadLine());

for (int i = 0; i < n; i++){

Console.Write("Enter the number: ");

int x = int.Parse(Console.ReadLine());

YesOrNo(x, n);

}

}

}

**Листинг программы на Java**

import java.util.Scanner;

public class TimeChecker {

public static boolean WhatTimeIsIt(int number) {

int hours = number / 100;

int minutes = number % 100;

return (hours >= 0 && hours <= 23) && (minutes >= 0 && minutes <= 59);

}

public static void YesOrNo(int number, int n) {

if (WhatTimeIsIt(number)) {

System.out.println("YES");

} else {

System.out.println("NO");

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("n = ");

int n = scanner.nextInt();

for (int i = 0; i < n; i++) {

System.out.print("Enter the number: ");

int x = scanner.nextInt();

YesOrNo(x, n);

}

}

}

**Листинг программы на Ruby**

def what\_time\_is\_it(number)

hours = number / 100

minutes = number % 100

(hours >= 0 && hours <= 23) && (minutes >= 0 && minutes <= 59)

end

def yes\_or\_no(number, n)

if what\_time\_is\_it(number)

puts "YES"

else

puts "NO"

end

end

print "n = "

n = gets.chomp.to\_i

n.times do

print "Enter the number: "

x = gets.chomp.to\_i

yes\_or\_no(x, n)

end

**Листинг программы на Rust**

use std::io;

fn what\_time\_is\_it(number: i32) -> bool {

let hours = number / 100;

let minutes = number % 100;

(hours >= 0 && hours <= 23) && (minutes >= 0 && minutes <= 59)

}

fn yes\_or\_no(number: i32, n: i32) {

if what\_time\_is\_it(number) {

println!("YES");

} else {

println!("NO");

}

}

fn main() {

let mut input = String::new();

println!("n = ");

io::stdin().read\_line(&mut input).expect("Enter only int!");

let n: i32 = input.trim().parse().expect("Enter only int!");

for \_ in 0..n {

let mut input = String::new();

println!("Enter the number: ");

io::stdin().read\_line(&mut input).expect("Enter only int!");

let x: i32 = input.trim().parse().expect("Enter only int!");

yes\_or\_no(x, n);

}

}

**Листинг программы на Kotlin**

import java.util.Scanner

fun whatTimeIsIt(number: Int): Boolean {

val hours = number / 100

val minutes = number % 100

return (hours in 0..23) && (minutes in 0..59)

}

fun yesOrNo(number: Int, n: Int) {

if (whatTimeIsIt(number)) {

println("YES")

} else {

println("NO")

}

}

fun main() {

val scanner = Scanner(System.`in`)

print("n = ")

var n = 0

n = scanner.nextInt()

for (i in 0 until n) {

print("Enter the number: ")

var x = scanner.nextInt()

yesOrNo(x, n)

}

}

**Листинг программы на Haskell**

whatTimeIsIt :: Int -> Bool

whatTimeIsIt number = let

hours = number `div` 100

minutes = number `mod` 100

in (hours >= 0 && hours <= 23) && (minutes >= 0 && minutes <= 59)

yesOrNo :: [Int] -> Int -> IO ()

yesOrNo numbers n = mapM\_ checkTime numbers

where

checkTime number = if whatTimeIsIt number

then putStrLn "YES"

else putStrLn "NO"

main :: IO ()

main = do

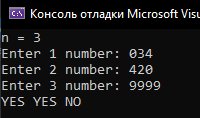
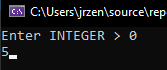
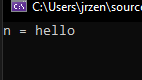
putStrLn "n = "

n <- readLn

numbers <- sequence [readLn | \_ <- [1..n]]

yesOrNo numbers n

**Результат работы программы**

** **

**Вывод**: изучили функции ввода-вывода данных, программирования вычисления значения выражения. Реализовали линейный вычислительный процесс на следующих ЯП согласно варианту: JavaScript, Go (Golang), Swift, С#, Haskell, Ruby, Rust, C++, Python, Java, Kotlin.

[al1ccce/LabWork2: literally LabWork2 (github.com)](https://github.com/al1ccce/LabWork2)