# Министерство образования Республики Беларусь

УПРАВЛЕНИЕ ОБРАЗОВАНИЯ МОГИЛЕВСКОГО ОБЛАСТНОГО ИСПОЛНИТЕЛЬНОГО КОМИТЕТА

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«Могилевский государственный политехнический колледж»

# Домашняя контрольная работа №1

# По дисциплине: «Основы алгоритмизации и программирования»

Группа ПО-455

## Выполнил ФИО

Шифр 21

# 2020

1. **Составьте программу заполнения одномерного массива из n элементов так, чтобы первый элемент был равен 1, второй ‑ 2, а каждый следующий был равен сумме двух предыдущих.**

**Алгоритм решения задачи:**

Начало

Ввод N

A[1]:=1

A[2]:=2

i от 3 до N

A[i]:=A[i-1]+A[i-2]

i от 1 до N

Вывод A[i]

Конец

**Текст программы:**

program Fibonachi5;

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

var

N: integer;

i : integer;

A:array[1..100] of integer;

begin

Write('Введите кол-во элементов массива: ');

ReadLn(N);

A[1]:=1;

A[2]:=2;

for i := 3 to N do {Вычисляем оставшиеся элементы массива}

begin

A[i]:=A[i-1]+A[i-2];

end;

for i:= 1 to N do{По всем элементам массива выводим на экран}

begin

Write(A[i],’ ’)

end;

WriteLn;

Writeln('Нажмите Ввод');

Readln;

end.

**Тест:**

Введите кол-во элементов массива: 15

1 2 3 5 8 13 21 34 55 89 144 233 377 610 987

Нажмите Ввод

1. **Дана квадратная матрица А(4,4). За один просмотр элементов матрицы А(4,4) сформируйте вектор С(4), каждый j-й элемент которого равен произведению элементов j-го столбца исходной матрицы, и вектор D(4), каждый j-й элемент которого равен сумме соответствующей строки матрицы А.**

**Алгоритм решения задачи:**

Начало

Ввод l,p

Nkrat=0

Notr=0

I от 1 до l

J от 1 до p

Ввод B[l,p]

B[l,p] mod 3=0

B[l,p]<0

Да

Нет

Да

Нет

Nkrat:=Nkrat+1

Notr:=Notr+1

Вывод Nkrat, Notr

Конец

**Текст программы:**

program Project60;

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

var

Nkrat,Notr:integer;

l,p:integer;

i,j:integer;

B:array[1..100,1..100] of integer;

begin

Write('Введите l: ');{Указываем размерность массива}

ReadLn(l);

Write('Введите p: ');

ReadLn(p);

Nkrat:=0;

Notr:=0;

for I := 1 to l do{Вводим элементы массива}

begin

for J := 1 to p do

begin

Write('Введите B[',i,',',j,']: ');

ReadLn(B[i,j]);

if (B[i,j] mod 3 = 0) then inc(Nkrat);{Делится ли элемент на 3?}

if (B[i,j]<0) then inc(Notr);{Отрицательный ли элемент?}

end;

Writeln;

end;

{ Выводим результаты подсчетов на экран}

Writeln('Количество кратных 3 элементов: ',Nkrat);

Writeln('Отрицательных элементов: ',Notr);

Writeln('Нажмите Enter');

Readln;

Readln;

end.

**Тест:**

Введите l: 3

Введите p: 4

Введите B[1,1]: 1

Введите B[1,2]: 2

Введите B[1,3]: 3

Введите B[1,4]: 4

Введите B[2,1]: -6

Введите B[2,2]: -5

Введите B[2,3]: -8

Введите B[2,4]: -9

Введите B[3,1]: -11

Введите B[3,2]: 12

Введите B[3,3]: -89

Введите B[3,4]: 10

Количество кратных 3 элементов: 4

Отрицательных элементов: 6

1. **Напишите программу вычисления суммы факториалов всех нечетных чисел от 1 до 55, используя подпрограмму вычисления факториала.**

**Алгоритм подпрограммы вычисления факториала:**

Начало, N

R:=1

i от 1 до n

R:=R\*i

Конец, R

**Алгоритм решения задачи:**

Начало

Ввод N, M

Да

Нет

N>M

Вывод сообщения об ошибке

С:=fact(N)/(fact(M)\*(N-M))

Вывод C

Конец

**Текст программы:**

program Project70;

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

var

c:int64;

n,m:integer;

/////////////////////////подпрограмма нахождения факториала

function fact(n:integer):int64;

var

i:integer;

r:int64;{для большей вместимости}

begin

r:=1;

for i := 1 to n do{перебираем подряд числа до N}

r:=r\*i;{поочерёдно домножаем}

result:=r;{Возвращаем результат}

end;

begin

Write('Введите N: ');{Вводим исходные данные}

ReadLn(n);

Write('Введите M: ');

ReadLn(m);

if N>M then{если выполнено условия неравенства}

begin

c:=fact(n) div (fact(m)\*(n-m));

{DIV тут чтобы не прибегать к типу с плавающей точкой, результат всё равно будет гарантированно целочисленным}

Writeln('С=',c);{Выводим результат}

end

else

Writeln('Не выполнено условие - N>M'); );{Выводим сообщение ошибки}

WriteLn('Нажмите Enter');

ReadLn;

end.

**Тест:**

Введите N: 6

Введите M: 4

С=15

Нажмите Enter

**Тест при ошибке:**

Введите N: 10

Введите M: 20

Не выполнено условие - N>M

Нажмите Enter

**91 Введите строку, если она является записью римского числа – запишите ее как целое число.**

**Алгоритм решения задачи:**

Начало

Ввод sin

N:=0

I от 1 до length(sin)

Да

Нет

sin[i] in A

N:=N+1

sin[i]:=UpCase(sin[i])

Вывод sin, n

Конец

**Текст программы:**

**program** Romanian\_91;

**var**

s: string;

tI, tX: byte;

f: byte;

res: integer;

**begin**

**repeat**

WriteLn('Введите Z для окончания ввода');

Write('Введите число: ');

ReadLn(s);

tI := 0;

tX := 0;

res := 0;

// считаем сумму всех встретившихся

// римских чисел согласно их значениям

**for** f := 1 **to** length(s) **do**

**begin**

**case** s[f] **of**

'I': inc(res, 1);

'V': inc(res, 5);

'X': inc(res, 10);

'L': inc(res, 50);

'C': inc(res, 100);

**end**;

// res - общая сумма чисел, но нужно обработать комбинации

// вида I перед X или V

**if** (s[f] = 'V') **and** (tI <> 0) **then**

**begin**// ситуация когда I перед V

tX := 0;

res := res - 2 \* tI;//удваиваем потому что уже первый раз прибавили к

// общей сумме а второй как действие модификатора

tI := 0;

**continue**// переходим к следующей итерации цикла без остальных

//проверок

**end**;

**if** (s[f] = 'X') **and** (tI <> 0) **then**

**begin**// ситуация когда I перед X

res := res - 2 \* tI;

tI := 0;

tX := 0;

**continue**

**end**;

// нужно обработать комбинации

// вида X перед L или C

**if** (s[f] = 'L') **and** (tX <> 0) **then**

**begin**// ситуация когда X перед L

res := res - tX \* 20;

tX := 0;

**continue**

**end**;

**if** (s[f] = 'C') **and** (tX <> 0) **then**

**begin**// ситуация когда X перед C

res := res - tX \* 20;

tX := 0;

**continue**

**end**;

// запоминаем количество I и X вдруг они потом

// станут модификаторами для X L C V

**if** s[f] = 'I' **then**

**begin**

inc(tI);// считаем количество I вдруг потом будет V или X

**continue**

**end**;

**if** s[f] = 'X' **then**

**begin**

inc(tX);// считаем количество X вдруг потом будет C или L

**continue**

**end**

**end**;

WriteLn('Ответ -> ', res);

**until** (s = 'Z') **or** (s = 'z')// в любом регистре

**end**.

**138 Дано: Х1={2,4,6,8,10}; Х2={1,3,5Д7}.- Напишите программу, которая формирует множество Y=(XlX2ХЗ) и выделите из него подмножество Y1, которое представляет все цифры, входящие BY.**

**Алгоритм решения задачи:**

Начало

Y :=(X1\*X2)+(X1\*X3)+(X2\*X3)

Вывод «Множество Y=»

I от 1 до 30

Да

Нет

i in Y

Вывод i

I от 1 до 30

Да

Нет

(i in Y)and(I mod 9=0)

Include(Y1,i)

I от 1 до 30

(i in Y1)

Да

Нет

N:=N+1

Вывод N

Конец

**Текст программы:**

program Project138;

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

Const{Задаем множества}

X1:set of byte=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20];

X2:set of byte=[10,20,30];

X3:set of byte=[1,3,5,7,9,11,13,15,17,19,21];

var

Y1,Y:set of byte;

i,n:byte;

begin

Y :=(X1\*X2)+(X1\*X3)+(X2\*X3);{Вычисляем Y1}

Writeln('Множество Y=');

for I := 1 to 30 do{Выводим множество Y}

if I in Y then

Write(I,' ');

Writeln('');

for I := 1 to 30 do{Заполняем Y1 элементами Y кратными 9}

if (I in Y)and(I mod 9 = 0) then

Include(Y1,i);

N:=0;

for I := 1 to 30 do{Подсчитываем мощность Y1}

if I in Y1 then

N:=N+1;

{Выводим результаты}

WriteLn('Мощность множества Y1=',N);

WriteLn('Нажмите Enter');

ReadLn;

end.

**Тест:**

Множество Y=

1 3 5 7 9 10 11 13 15 17 19 20

Мощность множества Y1=1

Нажмите Enter

**Список использованных источников**

1. ГОСТ ИСО/ МЭК 2382-99. Информационные технологии. Словарь. Ч. 1. Основные термины.
2. ГОСТ19.701-90 (ИСО 5807-85). Описание символов. Правила применения символов и выполнения схем.
3. СТУ СМК 01-32-2019. Стандарт учреждения. Общие требования к оформлению текстовых документов».
4. Макаров, В.Л. Программирование и основы алгоритмизации / В.Л. Макаров. – Санкт-Петербург: 2003.
5. Котов, В.М. Информатика. Методы алгоритмизации / В.М. Котов. – Нар. асвета: 2000.
6. Зубок, Д.А. Основы программирования в среде TURBO PASCAL / Д.А Зубок, С.В. Краснов, А.В. Маятин. – Санкт-Петербург: 2009.
7. Беляева, И.В. Основы программирования на языке TURBO PASCAL / И.В. Беляева. – Ульяновск: 2011.
8. Красиков, И.В. Алгоритмы. Просто как дважды два / И.В. Красиков. – М.: Эксмо, 2007.
9. Окулов С.М. Программирование в алгоритмах / С.М. Окулов. – М.: БИНОМ. Лабор. знаний, 2002.
10. Никлаус Вирт Алгоритмы и структуры данных / Вирт Никлаус. – Москва, 2010.