# Министерство образования Республики Беларусь

УПРАВЛЕНИЕ ОБРАЗОВАНИЯ МОГИЛЕВСКОГО ОБЛАСТНОГО ИСПОЛНИТЕЛЬНОГО КОМИТЕТА

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«Могилевский государственный политехнический колледж»

# Домашняя контрольная работа №1

# По дисциплине: «Структуры и алгоритмы обработки данных»

Группа ПО-370

## Выполнил С. В. Микулич

Шифр 19

# 2018

**22 Определите наибольшее количество идущих подряд положительных (отрицательных) элементов таблицы.**

**Алгоритм решения задачи:**

Начало

Ввод N

Ncurotr=0

Ncurpos=0

Ввод A[i]

I от 1 до N

I от 1 до N-1

(A[i]>0) and (A[i+1]>0)

Нет

Да

inc(Ncurpos)

Ncurpos:=1

Нет

Да

(A[i]<0) and (A[i+1]<0)

Ncurotr:=1

inc(Ncurotr)

Нет

Да

Ncurpos>Npos

Npos:=Ncurpos

Ncurotr>Notr

Да

Нет

Notr:=Ncurotr

Вывод Npos, Notr

Конец

**Текст программы:**

program Project22;

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

var

N, Npos, Notr : integer;

Ncurotr, Ncurpos : integer;

I : integer;

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

begin

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

ReadLn(N);

Ncurpos:=1;{Инициализируем переменные}

Ncurotr:=1;

for i := 1 to N do {Вводим элементы массива}

begin

Write('Введите A[',i,']: ');

ReadLn(A[i]);

end;

for I:= 1 to N-1 do{По всем элементам массива}

begin

if (A[i]>0) and (A[i+1]>0) then inc(Ncurpos)

else Ncurpos:=1;{Подсчитываем подряд положительные}

if (A[i]<0) and (A[i+1]<0) then inc(Ncurotr)

else Ncurotr:=1;{Подсчитываем подряд отрицательные}

if Ncurpos>Npos then Npos:=Ncurpos; {Выбираем большую последовательность если их несколько}

if Ncurotr>Notr then Notr:=Ncurotr; {Выбираем большую последовательность если их несколько}

end;

Writeln('Максимально подряд положительных элементов: ',Npos);

Writeln('Максимально подряд отрицательных элементов: ',Notr);

Writeln('Нажмите Enter');

Readln;

Readln;

end.

**Тест:**

Введите кол-во элементов массива: 10

Введите A[1]: 1

Введите A[2]: 2

Введите A[3]: 3

Введите A[4]: 4

Введите A[5]: -6

Введите A[6]: -5

Введите A[7]: -6

Введите A[8]: 2

Введите A[9]: 3

Введите A[10]: 1

Максимально подряд положительных элементов: 4

Максимально подряд отрицательных элементов: 3

**60 Дана матрица В(l,p). Определите сумму элементов, кратных 3, и количество отрицательных элементов.**

**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

**70 Определите число сочетаний из n по m (n>m), по формуле С=n!/m!(n-m).**

**Алгоритм подпрограммы вычисления факториала:**

Начало, N

R:=0

I от 1 до n

R:=R\*1

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

**113 В заданном тексте предложения замените строчные буквы на прописные и подсчитайте количество произведенных замен**

**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 sout, n

Конец

**Текст программы:**

program Project113;

{ВНИМАНИЕ - обрабатывает Английскую раскладку}

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

const

A:set of char = ['a','b','c','d','e','f','g','h','i',{набор строчных букв}

'j','k','l','m','n','o','p','q',

'r','s','t','u','v','w','x','y','z'];

var

sin:string;

i,n:integer;

begin

Write('Введите строку: ');{строка для обработки}

ReadLn(sin);

for I := 1 to length(sin) do{цикл по буквам строки}

begin

if sin[i] in A then{проверка на строчную букву}

begin

n:=n+1;{увеличиваем счетчик замен}

sin[i]:=upcase(sin[i]);{Преобразуем в верхний регистр}

end;

end;

Writeln('Результирующая строка: ',sшт);

Writeln('Количество замен: ',n);

WriteLn('Нажмите Enter');

ReadLn;

end.

**Тест:**

Введите строку: AavvvNMJnnasUyUy

Результирующая строка: AAVVVNMJNNASUYUY

Количество замен: 10

Нажмите Enter

**131 Даны три множества X1= {1,2,3,...,20}, X2= {10,20,30,...,30} и X3= {1,3,5,...,19,21}. Сформируйте множество Y= (X1xX2)U(X1xX3)U(X2xX3), из которого выделите подмножество Y1 чисел, делящихся на 9 без остатка. На печать выведите множество Y и мощность множества Y1. Исходные множества введите с клавиатуры.**

**Алгоритм решения задачи:**

Начало

Ввод sin

N:=0

I от 1 до length(sin)

Да

Нет

sin[i] in A

N:=N+1

sin:=AnsiUppercase(sin)

Вывод sout, n

Конец

**Текст программы:**

program Project131;

{$APPTYPE CONSOLE}

{$R \*.res}

uses

System.SysUtils;

const

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);

Writeln('Множество Y=');

for I := 1 to 30 do

if I in Y then

Write(I,' ');

Writeln('');

for I := 1 to 30 do

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

Include(Y,i);

N:=0;

for I := 1 to 30 do

if I in Y1 then

N:=N+1;

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=0

Нажмите Enter

**Список используемых 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.