## Реферат з інформатики

**Програмування: масиви та рядки.**

 1. Одновимірні масиви

 Масив у програмуванні – це тип структури даних, що має складені значення.

Якщо треба об'єднати кілька сотень або тисяч значень,що мають один тип, нам потрібні структури, в яких змінні однотипні й відрізняються номерами.

 Масив – це змінна, утворена послідовністю змінних, причому:

 усі вони (компоненти, або елементи масиву) мають той самий тип;

 кожний компонент має свій номер у послідовності (індекс) і відрізняється ним від інших елементів(ідентифікується);

 множина індексів (індексова множина) скінченна й зафіксована в означенні масиву та в процесі виконання програми не змінюється;

 можливість обробки компонента, або його доступність, не залежить від його місця в послідовності (елементи рівнодоступні).

Кількість елементів індексової множини називається довжиною масиву.

Подивимося на масив із точки зору математики. Нехай компоненти масиву мають тип T, а індекси – тип I.Значенням змінної-масиву є послідовність значень типу T, занумерованих значеннями типу I. Множина всіх таких функцій утворює носій для типу, який у мові Паскаль означається виразом вигляду

 *array [I ] of T.*

Наприклад, масив, у якому треба зберігати коефіцієнти полінома, міг би мати тип array[0..101]of real. У такому масиві 102 компоненти дійсного типу із номерами від 0 до 101. Або масив, у якому треба зберігати кількості символів, прочитаних десь, міг би мати тип array [ char ] of integer. У ньому 256 цілих змінних, а їх номерами є символи.

Типом компонентів може бути довільний тип, окрім файлів. Типом індексів I – будь-який

перелічуваний тип. Щоправда, система Бор ланд-Паскаль не дозволяє вказувати типи integer та word, а тим паче тип longint, як типи індексів. Там занадто багато елементів. Але це не страшно, оскільки система дозволяє створювати масиви навіть із ще більшою кількістю елементів.

Якщо тип індексів означається виразом у дужках [ ] як діапазон, то, зрозуміло, там пишуться дві сталі, і ніяких змінних там не може бути.

В означенні масивів як змінних немає ніяких особливостей. Наприклад, ми можемо написати як

 *type ART=array[0..101]of real; var A : ART;*

так і

*var A : array[0..101]of real;*

В обох випадках змінна A складається зі 102 дійсних змінних. Вони ідентифікуються виразами A[0], A[1], … ,A[101]. Або виразами вигляду A[індексовий-вираз], де індексовий-вираз має значення від 0 до 101.

. Нехай ім'я означено як масив типу array[I] of T, E – вираз типу I. Тоді вираз ім'я[E] задає елемент цього масиву, тобто змінну типу T, номер якої в масиві є значенням виразу E.

Вирази з операцією [] допустимі в операторах скрізь, де вживається змінна відповідного типу, за винятком того, що елемент масиву не може бути параметром циклу. Наприклад, якщо діє означення типу ART та змінної A, то можна писати щось на зразок

 *A[1] :=1; A[2] := A[1]+2;*

 *for k := 3 to 101 do readln(A[k]);*

 *writeln(A[1]+A[2]+A[3]+A[4]).*

Індексування – це єдина операція над масивами як цілісними об'єктами. Тому обробка масивів описується через обробку їх компонентів. Щоправда, в мові Бор ланд-Паскаль можна присвоювати однотипні масиви. Наприклад, за дії означень

*var a, b : array[char] of integer; ch : char;*

можна замість циклу

*for ch:= chr(0) to chr(255) do b[ch]:=a[ch*];

написати лаконічно

*b := a;*

Задачі

1)На вхід програми подається N цілих чисел X1, ... , XN із діапазону 0 ..100; N>0. Написати програму обчислення:

- їх середнього арифметичного значення M та дисперсії D, тобто середнього арифметичного квадратів різниць між числами та M:

D = ( ( X1 - M )2 + … + ( XN - M )2 ) / N;

- кількостей повторень K1, K2, … , K100 кожного з чисел 1, 2, … , 100.

Числа надходять у програму

а) від стандартного пристрою введення, тобто клавіатури;

б) від генератора випадкових чисел.

2). Черга – це така послідовність, що елементи додаються в її кінець, а вилучаються з її початку. Написати програму роботи з чергою цілих, поданою в масиві.Потрібно забезпечити:

- скидання черги (вилучення всіх її елементів);

- обчислення кількості елементів у черзі;

- перевірки, чи порожня черга;

- обчислення обсягу вільного місця в черзі (кількість елементів, які можна додати до її заповнення);

- додавання елемента в кінець черги;

- добування та вилучення елемента з її початку.

3).Стек, або магазин – це така послідовність, що елементи додаються й вилучаються з її початку. Написати програму роботи зі стеком цілих, поданим у масиві. Потрібно забезпечити:

- скидання стека (вилучення всіх його елементів);

- обчислення кількості елементів стека;

- перевірки, чи порожній стек;

- обчислення обсягу вільного місця в стеку (кількість елементів, які можна додати до його заповнення);

- додавання елемента до початку стека;

- добування й вилучення елемента з початку стека.

 2. Рядки

 Рядок у загальному значенні цього слова – скінченна послідовність символів. У програмуванні для подання рядків використовують масиви символів.

 Типи рядків мають свої специфічні операції, не означені над масивами символів, тобто рядки та символьні масиви є цілком різними типами.

Змінна-рядок є масивом символів разом із додатковим компонентом. В означенні типу задається довжина n послідовності символьних компонентів, індексованих цілими 1,…,n. Додатковий компонент указує довжину послідовності символів, поданої в масиві. Ця довжина значення-рядка може змінюватися в процесі виконання програми від 0 до довжини масиву n.

У мові Бор ланд-Паскаль тип рядків задається виразом вигляду

 *string[n],*

де n – ціла стала з діапазону 1..255 (можливо, іменована). Наприклад, string[80]. Змінна такого типу є масивом символів із індексами від 0 до n. Значення-рядок подається змінними з індексами від 1 до n, змінна з індексом 0подає довжину рядка. Точніше, якщо її значення c, то m=ord(c) розглядається як довжина значення-рядка. У процесі виконання програми вона може мінятися від 0 до n.

Елементи масиву з індексами від m+1 до n недоступні; їх значення можна розглядати як "сміття". Спроба присвоїти щось цим елементам або взяти їх значення призведе до аварійного завершення програми.

Оскільки невід'ємна довжина значення-рядка подається в одному байті, вона не може перевищувати 255. Звідси "маємо, те що маємо", тобто обмеження 255 на довжини рядків. Ім'я типу string еквівалентне виразові string[255].

Найпростішими виразами типу рядок є ім'я змінної-рядка, рядкова стала (літерал), або вираз типу char. Над рядками означена бінарна операція катенації (або дописування); її знаком є '+'. Результат одержується дописуванням правого операнда до лівого: значенням виразу '123'+'45' є '12345'. Ціла довжина рядка повертається з виклику функції length із аргументом-рядком: length('123') = 3.

Рядки – єдиний нескалярний тип, змінні та вирази якого можуть бути аргументами стандартних процедур читання та запису. Особливості виконання цих процедур розглядаються пізніше. Рядки також можуть повертатися як значення функцій.

Вираз типу "рядок" можна присвоїти змінній-рядку. Його символи присвоюються елементам змінної, починаючи з першого. Якщо довжина значення більша максимально можливої довжини n змінної, то присвоюються лише n перших символів. При цьому довжина значення (або n) неявно присвоюється додатковому компоненту змінної-рядка (як символ нульовому елементу масиву). Наприклад, нехай змінна s означена як string[3]. Після присвоювання

*s:='12345'*

чотири її компоненти з індексами 0, 1, 2, 3 матимуть значення chr(3), '1', '2', '3', а після

*s:='12'*

– chr(2), '1', '2', а останній компонент буде недоступним, і його значення буде "сміттям". За останнього значення змінної s виконання оператора

*s:=s+'9'*

надасть їй значення '129', а оператора

*s:='9'+s*

– значення '912'. В обох випадках s[0]=chr(3). Після присвоювання s:='' змінна s подає порожній рядок:

s[0]=chr(0), а решта елементів недоступні.

Змінити довжину значення можна явно, змінивши нульовий символ. Якщо після s:='' виконати s[0]:=2, то значення компонентів із індексами 1 і 2 зі "сміття" перетворяться на значення-рядок. Присвоювання іншим елементам рядка не змінює довжини його значення.

У типах рядків означено операції порівняння =, <>, <, … . За означенням, рядки рівні, якщо мають ту саму довжину, і в її межах відповідні символи однакові. У противному разі вони не рівні. Упорядкування рядків залежить від системи програмування.

У системі програмування Бор ланд-Паскаль означено багато корисних підпрограм обробки рядків.

Розглянемо лише чотири з них.

Функція з заголовком

*function copy ( s : string; ind, cnt : byte ) : string*

задає повернення підрядка рядка s, що починається з s[ind] і має довжину cnt. Наприклад, copy('abcd', 2, 2)='bc'.

Функція з заголовком

*function pos ( subs, s : string ) : byte*

задає повернення номера того елемента в рядку s, починаючи з якого subs входить у s як підрядок (якщо не входить, то повертається 0). Наприклад, pos('bc', 'abcd')=2, pos('aa', 'abcd')=0.

Процедура з заголовком

*procedure val(s : string, var v; var ErrCode : integer)*

задає перетворення зображення числа в рядку s у числовий тип і присвоювання його змінній v. Якщо перетворення дійсно можливе, то значенням ErrCode буде 0. У противному разі її значенням буде позиція з символом у рядку, починаючи з якого перетворення неможливе. Тип аргументу, відповідного параметрові v,повинен мати тип, відповідний змісту рядка s. Так само зміст рядка повинен задавати число, представне в типі цього аргументу. Наприклад, за s='1.3' або s='1E2' другий аргумент повинен бути дійсного типу, а не цілого.

Аналогічно за його типу integer у рядка не повинно бути значень, що подають числа, більші 32767 або менші -32768.

Процедура з заголовком

*procedure delete(var s : string; start, len : integer)*

задає знищення len символів, починаючи з позиції start у рядку s. Наприклад, за s='abcdef' після виклику delete(s,3, 3) рядок s матиме значення 'abf'. За start=0 або len=0 або start>length(s) рядок не змінюється. За start+len>length(s) з s вилучається підрядок до кінця рядка.

Задачі

1).Що друкується в результаті виконання програми:

а) program strconc ( input, output );

var a: integer; c: char; s: string;

begin

s :='';

for a := 0 to 2 do

begin

c := chr ( ord ( '0' ) + a ); s := c + s + c; writeln ( s )

end

end.

б) program concstr ( input, output );

var a: integer; c: char; s: string;

begin

s :=''; c:= chr ( 47 );

while length ( s ) < 7 do

begin

c := succ ( c ); s := s + c + s; writeln ( s )

end

end.

 3. Матриці та багатовимірні масиви

Розглянемо прямокутну таблицю з m\* n однотипиних елементів як послідовність із m рядків, у кожному з яких n елементів. Послідовності певної довжини подаються в мовах програмування масивами. Отже, виникає поняття "масив, елементами якого є масиви", або двовимірний масив. Якщо елементи прямокутної таблиці самі є послідовностями або таблиці утворюють послідовність певної довжини, то виникає поняття тривимірного

масиву тощо.

Означення багатовимірних масивів та зображення їх елементів у мові Паскаль опишемо за допомогою простого прикладу. Позиція в грі "хрестики-нулики на полі 3\* 3" подається квадратною таблицею з символів 'x', '0' або ' '(пропуск). Пронумеруємо клітинки поля, як у шахах – літерами 'a', 'b', 'c' по горизонталі та числами 1, 2, 3 по вертикалі. Тоді рядки таблиці можна подати масивами типу

*type Row = array [ 'a' .. 'c' ] of char;*

Таблицю можна розглянути як послідовність трьох рядків і подати масивом типу

*type Table = array [ 1 .. 3 ] of Row;*

Партія, тобто послідовність позицій, має довжину не більше 9, і може подаватися масивом таблиць:

*type Game = array [ 1 .. 9 ] of Table;*

Масиви типу Table мають два виміри: номер рядка та номер символу в ньому; масиви типу Game – три: номери таблиці, рядка та символу. Вимір 1..9 у типі Game називається зовнішнім, вимір 'a'..'c', що нумерує символи в рядках, – внутрішнім.

Тип Game можна задати еквівалентним виразом, не означаючи імен типів Row і Table, а саме:

*type Game = array [ 1 .. 9 ] of*

*array [ 1 .. 3 ] of*

*array [ 'a' .. 'c' ] of char;*

Нехай A – змінна типу Game. Вираз вигляду A[ i ], де 1? i ? 9, задає змінну типу Table, або типу

*array [ 1 .. 3 ] of array [ 'a' .. 'c' ] of char;*

вираз вигляду A[i][j], де 1? j? 3, – змінну типу Row, або типу

*array[ 'a' .. 'c' ] of char;*

вираз вигляду A[i][j][k], де 'a' ? k ? 'c', – змінну типу char.

Мова Паскаль допускає іншу форму задання типів та елементів багатовимірних масивів: виміри та індекси записуються через кому в спільних дужках. Так, означення

*type Game1 = array [ 1 .. 9, 1 .. 3, 'a' .. 'c' ] of char*

еквівалентне означенню типу Game, а вираз A[i, j, k] – виразові A[i][j][k].

Елементи багатовимірних масивів розташовуються в пам'яті послідовно, найшвидше в них змінюється внутрішній індекс, найповільніше – зовнішній. Зокрема, двовимірні масиви (матриці) розташовуються за рядками. Так, послідовні числа масиву типу

*array [ 1 ..2, 'a' ..'b' ] of real*

мають набори індексів [1, 'a'], [1, 'b'], [2, 'a'], [2, 'b'], а послідовні символи в масиві типу Game – [1, 1, 'a'], [1, 1, 'b'], [1, 1, 'c'], [1, 2, 'a'], … , [1, 3, 'c'], [2, 1, 'a'], … , [9, 3, 'c'].

Задачі

1).У матриці розмірами M \* N обміняти місцями

а) два рядки, б) два стовпці, задані номерами.

2)Транспонувати квадратну матрицю без використання додаткової матриці.

3)Квадратну матрицю повернути за годинниковою стрілкою на

а) 90° ; б) 180° . Додаткову матрицю не використовувати.

4). Елемент матриці називається сідловим, якщо його значення є мінімальним у рядку й максимальним у стовпці, на перетині яких він знаходиться (або навпаки, максимальним у рядку й мінімальним у стовпці). Написати процедуру повернення номерів рядка та стовпця якого-небудь із сідлових елементів (якщо таких немає, то повертається пара номерів зовні індексної множини матриці).

5).Написати процедуру обчислення добутку двох матриць.

6). Елементи N-вимірного масиву розмірів M1 ? … ? MN розміщаються в пам'яті комп'ютера послідовно так, що найшвидше змінюється їх останній індекс, найповільніше – перший. Написати функцію обчислення лінійного індексу елемента (його номера в порядку розташування в пам'яті) за заданими розмірами M1, … , MN та індексами елемента в N-вимірному масиві. Написати процедуру обчислення індексів

елемента в багатовимірному масиві за його лінійним індексом та розмірами M1, … , MN. Значенням N є:

а) 2; б) 3; в) 4.