

Процедурне програмування

Виконайте ОДНУ (на Ваш вибір) лабораторну роботу мовою програмування (на Ваш вибір)

На Ваш вибір – Pascal або C.

Виберіть будь-яку лабораторну роботу для ознайомлення з "процедурним програмуванням". Виконайте оцінку складності алгоритмів, що використовуються.

[Pascal 1](#)

[Pascal 2](#)

Програмування з використанням мови C

[Робота C1](#)

[Робота C2](#)

Найпростіший IDE для роботи з мовою C можна "взяти" тут: [IDE CodeBlocks](#)

Перейдіть за посиланням <http://www.codeblocks.org/downloads/binaries/> скачайте та встановіть CodeBlocks обов'язково з компіляторами MINGW

Рекомендоване посилання [codeblocks-20.03mingw-32bit-setup.exe](#)

Для Паскаля можна рекомендувати [PascalABC.NET - Windows, Linux](#)

Pascal робота № 1

Використання процедур та функцій

1. Обчислити значення виразу (використовувати функцію)

$$\frac{3(\sin \sqrt{x} + x) + 5}{1/2(\sin \sqrt{z^3 + z^3})} - \sin\left(\frac{y}{2}\right) - \frac{y}{2} - 8$$

2. За заданими 10-елементними масивами А, В і С обчислити:

$$t = \begin{cases} \frac{\min(b_i)}{\max(a_i)} + \frac{\max(c_i)}{\min(b_i + c_i)}, & \text{при } \min(a_i) < \max(b_i) \\ \frac{\max(b_i + c_i)}{\max(b_i + c_i) + \min(c_i)}, & \text{при } \min(a_i) \geq \max(b_i) \end{cases}$$

3. Задано три матриці А(3,3), В(2,2), С(3,3). Знайти мінімальне із трьох чисел х, у, z, де х - слід матриці А, у- слід матриці В, z- слід матриці С. Услід матриці називається сума елементів головної діагоналі. Обчислення сліду оформити як процедури.

4. Обчислити значення інтеграла методом трапецій (використовувати функцію)

$$\int_{0.1}^1 \sqrt{e^x - 1} dx$$

5. Дано довжини а, b, с сторін деякого трикутника. Знайти медіани трикутника, сторонами якого є медіани вихідного трикутника.
6. Дано дві квадратні речові матриці. Надрукувати квадрат тієї їх, у якій найменший слід (сума діагональних елементів), вважаючи, що така матриця одна.
7. За заданими 10-елементними цілими масивами Х і Y обчислити

$$u = \begin{cases} \sum_{i=1}^{10} x_i^2, & \text{при } \sum_{i=1}^5 x_i y_i > 0 \\ \sum_{i=1}^{10} y_i^2, & \text{при } \sum_{i=1}^5 x_i y_i \leq 0 \end{cases}$$

8. Дано коефіцієнти многочленів Р(х) і Q(х) 5-го ступеня і дано речовинне а. Обчислити Р(а+Q(а)Р(а+1)).
9. За дійсним числом а>0 обчислити величину

$$\frac{\sqrt[3]{a} - \sqrt[6]{a^2 + 1}}{1 + \sqrt[7]{3 + a}}$$

10. Обчислити значення інтеграла методом трапецій (використовувати функцію)

$$\int_2^{2.5} \frac{dx}{x \ln^2 x}$$

11. Скласти програму знаходження сум елементів кожного стовпця матриць А і В, використовуючи процедуру.
12. Задано три одновимірні масиви X, Y, Z. Упорядкувати за зростанням три числа a, b і c, де a - мінімальний елемент масиву X, b - мінімальний елемент масиву Y, а c - мінімальний елемент масиву Z. Пошук мінімального елемента оформити у вигляді процедури.
13. Обчислити значення вираз (використовувати функцію)

$$\left(\arccos x - \sqrt{1 - 0.3x^2} \right) \left(\arccos \frac{y}{2} - \sqrt{1 - \frac{0.3y^2}{4}} + 5 \right)$$

14. Обчислити значення певного інтегралу методом Сімпсона

$$\int_1^{1.8} \ln x dx$$

15. Скласти програму для обчислення значення функції $z = (x_1 + x_2) / (y_1 + y_2)$, де

$$x_1, x_2 - \text{коріння рівняння } ax^2 + bx + c = 0;$$

$$y_1, y_2 - \text{коріння рівняння } dy^2 + ly + f = 0.$$

Для обчислення коренів рівняння використати процедуру.

16. Дано координати вершин двох трикутників. Визначити, який із них має велику площу.

17. Розв'язати рівняння $dx=c$ де d - довжина вектора $a = \{a_1, a_2, a_3, a_4\}$, c - довжина вектора $b = \{b_1, b_2, b_3\}$. Обчислення довжини вектора оформити як процедури.

18. Побудувати таблицю функції $y = cx^2 + d$ при x змінюється від 0 до 1 з кроком 0.1, де c - слід матриці A(3,3), d - слід матриці B(4,4). Слідом матриці називається сума елементів головної діагоналі. Обчислення сліду матриці оформити як процедури.

19. За заданими 10-елементними речовинними векторами x, y, z обчислити

$$w = \begin{cases} \prod_i (\sin(x_i) + 2), & \text{при } \prod_i (1 - y_i) > 0.5 \\ \prod_i (1 - z_i^2), & \text{при } \prod_i (1 - y_i) \leq 0.5 \end{cases}$$

20. Задано три матриці А, В, С. Упорядкувати за спаданням три числа x, y, z, де x - максимальний елемент матриці А, y - максимальний елемент матриці В, z - максимальний елемент матриці С. Пошук максимального елемента матриці оформити у вигляді процедури.

21. Скласти програму для обчислення значення функції $z = \sqrt{ab}$, де a - середнє арифметичне негативних елементів масиву $(x_1, x_2, \dots, x_{15})$, b - середнє арифметичне негативних елементів масиву $(y_1, y_2, \dots, y_{20})$. Для обчислення середнього арифметичного елемента масиву використовувати процедуру.

22. Задано два вектори $x = \{x_1, x_2, x_3, x_4\}$; $y = \{y_1, y_2, y_3, y_4\}$. Визначити кут φ між векторами x та y за формулою

$$\varphi = \arccos \frac{(x, y)}{\sqrt{(x, x)(y, y)}}$$

Обчислення скалярного твору оформити як процедури.

23. Скласти програму для обчислення

$$z = \frac{\sqrt{\sum_{i=1}^{10} \cos x_i} + \sqrt{\sum_{i=1}^{20} \sin y_i}}{\sqrt{\sum_{i=1}^{10} |x_i|}}$$

де X та Y задані масивами. Корінь квадратний із суми обчислити у процедурі.

24. Визначити, чи точка A(x,y) усередині трикутника $M_1M_2M_3$.

25. Обчислити значення інтеграла методом трапецій (використовуючи функцію)

$$\int_0^{1.7} x \operatorname{arctg} x \, dx$$

26. Скласти програму для обчислення значень функції

$$z_i = \frac{1}{s} \sum_{j=1}^5 x_{ij}, \quad s = \sum_{i=1}^{10} y_i,$$

x_{ij} - Елементи матриці 5×10 ;

y_i - Елементи масиву $(y_1, y_2, \dots, y_{10})$.

Підсумовування оформити процедурою.

27. Обчислити значення певного інтеграла методом Сімпсона

$$\int_1^e \frac{1 + \lg x}{x} \, dx$$

28. Задано чотири вектори

$$x = \{x_1, x_2, x_3\}, \quad y = \{y_1, y_2, y_3\}, \quad z = \{z_1, z_2, z_3, z_4\}, \quad p = \{p_1, p_2, p_3, p_4\}$$

Обчислити скалярний добуток векторів x і y порівняти його зі скалярним добутком векторів z і p знайти більше. Обчислення скалярного твору оформити як процедури.

29. Задано три квадратні рівняння

$$ax^2 + bx + c = 0, \quad dx^2 + fx + r = 0, \quad px^2 + qx + k = 0.$$

Знайти мінімальне значення серед коренів цих рівнянь. Рішення квадратного рівняння оформити як процедури.

30. Обчислити значення певного інтегралу методом Сімпсона

$$\int_0^{5.24} (1+x) \ln(1+x) dx$$

31. Сформувати речову матрицю $A = \{a_{ij}\}$ розмірності n де $a_{ij} = \frac{i!}{(j+1)!}$.

32. Скласти програму для обчислення функції

$$z = \frac{1}{45} \left(\sum_{i=1}^{10} \sin^2 a_i + \sum_{i=1}^5 \sin^2 b_i \right)$$

33. Обчислити вираз (використовувати функцію)

$$\frac{x - \frac{1}{3 + \sin 3.6x}}{z - \frac{1}{3 + \sin 3.6z}} \cos \left(2 - \frac{1}{3 + \sin 7.2} \right)$$

34. За заданими речовими числами c і d ($c < d$) обчислити

$$\int_c^d \arctg^2 x dx + \int_0^\pi \sin e^{10x} dx$$

Інтеграли обчислювати приблизно за формулою трапецій при $n=20$ для першого інтеграла і при $n=100$ для другого:

$$\int_a^b f(x) dx \approx h \left[f(a)/2 + \sum_{i=1}^{n-1} f(a+ih) + f(b)/2 \right]$$

$$\text{де } h = (b-a)/n.$$

35. Змоделювати процедуру `Ins`, аналогічну стандартній процедурі `Insert` роботи з рядками.

Pascal робота № 2

Програмування списків

Необхідно скласти програму роботи зі списком. Список необхідно розмістити в оперативній пам'яті. Список може бути пов'язаний у двох напрямках, тобто. допускати його перегляд з "голови" до "хвоста" та від "хвоста" до "голови". Як дані кожної ланки списку вибирається текстовий рядок.

Передбачити такі можливості програми:

- створення списку даних, які вводяться користувачем з консолі;
- створення списку з даних, які розміщені у довільному текстовому файлі;
- збереження даних списку довільний текстовий файл;
- виведення на консоль кількості ланок у списку;
- виведення на консоль даних із ланок списку у "прямому" та "зворотному" порядку із зазначенням номера ланки у списку;
- виведення на консоль даних ланки заданого номера;
- виведення на консоль номера ланки, що містить введені користувачем дані (пошук у списку) або повідомлення, що дані не знайдені;
- додавання нової ланки на початок або в кінець списку (дані про нову ланку вводяться з консолі);
- додавання нової ланки після ланки із зазначеним номером (дані про нову ланку вводяться з консолі);
- видалення ланки із зазначеним номером;
- тестовий роздрук ланок зі списку - роздруковуються дані та адреси.

Приклад тестової програми

```
Program test_spisok;
uses crt, objects;
  type spis=^zveno; {Структура ланки списку - список
однозв'язаний}
zveno=record
data:string; { дані ланки }
next:spis; {адреса наступної ланки}
end;

{ процедури моделювання однозв'язаного списку }

procedure sp_new (var head: spis; data: string); {Створити
список - список ПОРОЖНІЙ}
var curr: spis; {head - покажчик на ГОЛОВУ списку}
begin
new(curr);
curr^.data:=data;
curr^.next:=NIL;
head: = curr;
end;

procedure sp_add(head:spis; data:string); {додати до кінця
списку (head) нову ланку}
var z, curr: spis;
begin
z:=head;
```

```

while z^.next <> NIL do z:=z^.next;
new(curr);
z^.next:=curr;
curr^.data:=data;
curr^.next:=NIL;
end;

procedure sp_del (head: spis); {видалити список із пам'яті та
віддати пам'ять ОС}
var z, curr: spis;
begin
z:=head;
repeat
if z = NIL then break;
curr:=z;
z:=z^.next;
dispose(curr);
until z <> NIL;
writeln('spis is delete.....'); readln;
end;

procedure sp_print(head:spis); {роздруківка ланок списку}
var z: spis;
n: integer;
begin
z:=head; n:=0;
while z <> NIL do
begin
n:=n+1;
writeln('zap#', n, 'is', z^.data);
z:=z^.next;
end;
end;

procedure sp_file(head:spis; out:string); {збереження списку у
файл}
var z: spis;
n: integer;
oout: Text;
begin
{$I+}
assign (oout, out);
Rewrite(oout);
if IOResult <> 0 then
begin
writeln('Вказаний файл', out, 'не створено.....');
writeln('Список не збережено...'); exit;
end;
{$I-}
z:=head; n:=0;
while z <> NIL do
begin
n:=n+1;
writeln(oout, z^.data);
z:=z^.next;
end;
close (oout);
writeln(n, 'записів зі списку збережені у файл', out);
end;
{кінець процедури моделювання однозв'язаного списку}

{роздруківка 16 значень покажчиків Для ABC не потрібна}

```

```

function word2hex(w: Word):string;
const
hexChars: array [0..$F] of Char =
'0123456789ABCDEF';
begin
word2hex:=hexChars[Hi(w) shr 4] +
hexChars[Hi(w) and $F] +
hexChars[Lo(w) shr 4] +
hexChars[Lo(w) and $F] ;
end;
function pointer2string(p : pointer):string;
var zp: PtrRec;
begin
zp: = PtrRec (p);
pointer2string:=word2hex(zp.Seg) +
': ' +
word2hex(zp.Ofs) +
'$';
end; {роздруківка 16-х значень покажчиків}

procedure sp_print1(head:spis);
var z: spis;
n: integer;
begin
z:=head; n:=0;
while z <> NIL do
begin
n:=n+1;
write('add=', pointer2string(z) );
writeln(' zap# ', n, ' is:<=',
pointer2string(z^.next),
'>=', z^.data, '>' );
z:=z^.next;
end;
end;

{=====}
{Головна тест програма}

var inn: Text;
fname, buf: string;
sp1, sp2: spis;
begin
clrscr;
writeln('Створюємо список:');
write( 'введіть ім'я файлу або порожній рядок для створення списку з
консолі');
readln(fname);

if length(fname) = 0 then
begin {введення з консолі}
write('введіть перший запис:');
readln(buf);
якщо length(buf) = 0 then
begin
writeln('Список не створений....'); halt(5);
end;
sp_new(sp1, buf);
while true do {new zap in console}
begin
writeln('введіть новий запис (порожній рядок - закінчити):');
readln(buf); якщо length(buf) = 0 then break;
sp_add (sp1, buf);

```

```
end; {new zap in console}
end {введення з консолі}
else
begin {new zap in file}
  {$I+}
  assign (inn, fname);
  Reset(inn);
  if IOResult <> 0 then
  begin
  writeln('Вказаний файл не знайдено....');
  writeln('Список не створений....'); halt(5);
  end;
  {I-}
  if eof(inn) then
  begin
  writeln('Вказаний файл порожній....');
  writeln('Список не створений....'); halt(5);
  end;
  readln(inn, buf);
  sp_new(sp1, buf);
  while not eof(inn) do {new zap in file}
  begin
  readln(inn, buf);
  sp_add (sp1, buf);
  end; {new zap in file}
  close (inn);
  end; {new zap in file}

  writeln('sp1 створений.....');
  readln;
  writeln('вміст списку sp1:');

  sp_print(sp1);
  readln;
  writeln('вміст списку з адресами sp1:');

  sp_print1(sp1);

  readln;

  write('вказить ім'я файлу для збереження списку');
  readln(buf);
  sp_file(sp1, buf);

  sp_del (sp1);
end.
```

Робота № 1, мова програмування C

Застосування масивів та функцій

Ціль: Практика в організації ітераційних та арифметичних циклів. Здобуття навичок обробки масивів з використанням функцій. Організація динамічних масивів.

Постановка задачі

"Обчислення функцій з використанням їх розкладання в статечний ряд"

Ціль: Практика в організації ітераційних та арифметичних циклів, використання функцій.

Постановка задачі

Для змінюється від a до b з кроком $(b-a)/k$, де $(k=10)$, обчислити функцію $f(x)$, використовуючи її розкладання в статечний ряд у трьох випадках:

- для заданого n ;
- для заданої точності $\epsilon (\epsilon=0.0001)$;
- для "точного" значення (за аналітичною формулою).

Для порівняння знайти відносну похибку обчислення функції значення функції $\text{про_погр} = \text{ABS}(\text{точ_знач} - \text{наближ_знач}) / \text{точ_знач}$

Програма повинна містити три функції користувача, що виконують розрахунок значення Y трьома способами.

Варіанти

функція	діапазон зміни аргументу		сума
$y = 3^x$	$0,1 \leq x \leq 1$	0	$S = 1 + \frac{\ln 3}{1!} x + \frac{\ln^2 3}{2!} x^2 + \dots + \frac{\ln^n 3}{n!} x^n$
$y = -\ln \left 2 \sin \frac{x}{2} \right $	$\frac{\pi}{5} \leq x \leq \frac{9\pi}{5}$	0	$S = \cos x + \frac{\cos 2x}{2} + \dots + \frac{\cos nx}{n}$
$y = \sin X$	$0,1 \leq x \leq 1$	0	$S = x - \frac{x^3}{3!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$

	$y = X \operatorname{arctg} X - \ln \sqrt{1+x^2}$	$0,1 \leq x \leq 0,8$	0	$S = \frac{x^2}{2} - \frac{x^4}{12} + \dots + (-1)^{n+1} \frac{x^{2n}}{2n(2n-1)}$
	$y = e^x$	$1 \leq x \leq 2$	5	$S = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$
	$y = e^{x \cos \frac{\pi}{4}} \cdot \cos(x \sin \frac{\pi}{4})$	$0,1 \leq x \leq 1$	5	$S = 1 + \frac{\cos \frac{\pi}{4}}{1!} x + \dots + \frac{\cos n \frac{\pi}{4}}{n!} x^n$
	$y = \cos x$	$0,1 \leq x \leq 1$	0	$S = 1 - \frac{x^2}{2!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!}$
	$y = \frac{x \sin \frac{\pi}{4}}{1 - 2x \cos \frac{\pi}{4} + x^2}$	$0,1 \leq x \leq 0,8$	0	$S = x \sin \frac{\pi}{4} + x^2 \sin 2 \frac{\pi}{4} + \dots + x^n \sin n \frac{\pi}{4}$
	$y = \frac{1}{4} \ln \frac{1+x}{1-x} + \frac{1}{2} \operatorname{arctg} X$	$0,1 \leq x \leq 0,8$		$S = x + \frac{x^5}{5} + \dots + \frac{x^{4n+1}}{4n+1}$
0	$y = e^{\cos x} \cos(\sin x)$	$0,1 \leq x \leq 1$	0	$S = 1 + \frac{\cos x}{1!} + \dots + \frac{\cos nx}{n!}$
1	$y = (1+2x^2)e^{x^2}$	$0,1 \leq x \leq 1$	0	$S = 1 + 3x^2 + \dots + \frac{2n+1}{n!} x^{2n}$
2	$y = -\frac{1}{2} \ln(1 - 2x \cos \frac{\pi}{3} + x^2)$	$0,1 \leq x \leq 0,8$	5	$S = \frac{x \cos \frac{\pi}{3}}{1} + \frac{x^2 \cos 2 \frac{\pi}{3}}{2} + \dots + \frac{x^n \cos n \frac{\pi}{3}}{n}$
3	$y = \frac{1}{2} \ln x$	$0,2 \leq x \leq 1$	0	$S = \frac{x-1}{x+1} + \frac{1}{3} \left(\frac{x-1}{x+1}\right)^3 + \dots + \frac{1}{2n+1} \left(\frac{x-1}{x+1}\right)^{2n+1}$
4	$y = \frac{1}{4} \left(x^2 - \frac{\pi^2}{3}\right)$	$\frac{\pi}{5} \leq x \leq \pi$	0	$S = -\cos x + \frac{\cos 2x}{2^2} + \dots + (-1)^n \frac{\cos nx}{n^2}$
5	$y = \frac{1+x^2}{2} \operatorname{arctg} X$	$0,1 \leq x \leq 1$	0	$S = \frac{x^3}{3} - \frac{x^5}{15} + \dots + (-1)^{n+1} \frac{x^{2n+1}}{4n^2 - 1}$
6	$y = \frac{\pi^2}{8} - \frac{\pi}{4} x $	$\frac{\pi}{5} \leq x \leq \pi$	0	$S = \cos x + \frac{\cos 3x}{3^2} + \dots + \frac{\cos(2n-1)x}{(2n-1)^2}$
7	$y = \frac{e^x + e^{-x}}{2}$	$0,1 \leq x \leq 1$	0	$S = 1 + \frac{x^2}{2!} + \dots + \frac{x^{2n}}{(2n)!}$
8	$y = \frac{1}{2} - \frac{\pi}{4} \sin x $	$0,1 \leq x \leq 0,8$	0	$S = \frac{\cos 2x}{3} + \frac{\cos 4x}{15} + \dots + \frac{\cos 2nx}{4n^2 - 1}$
9	$y = e^{2x}$	$0,1 \leq x \leq 1$	0	$S = 1 + \frac{2x}{1!} + \dots + \frac{(2x)^n}{n!}$

0	$y = \left(\frac{x^2}{4} + \frac{x}{2} + 1\right)e^{x/2}$	$0,1 \leq x \leq 1$	0	$S = 1 + 2\frac{x}{2} + \dots + \frac{n^2 + 1}{n!} \left(\frac{x}{2}\right)^n$
1	$y = \operatorname{arctg} X$	$0,1 \leq x \leq 1$	0	$S = x - \frac{x^3}{3} + \dots + (-1)^n \frac{x^{2n+1}}{2n+1}$
2	$y = \left(1 - \frac{x^2}{2}\right) \cos x - \frac{x}{2} \sin x$	$0,1 \leq x \leq 1$	5	$S = 1 - \frac{3}{2}x^2 + \dots + (-1)^n \frac{2n^2 + 1}{(2n)!} x^{2n}$
3	$y = 2(\cos^2 x - 1)$	$0,1 \leq x \leq 1$	5	$S = -\frac{(2x)^2}{2} + \frac{(2x)^4}{24} + \dots + (-1)^n \frac{(2x)^{2n}}{(2n)!}$
4	$y = \ln\left(\frac{1}{2 + 2x + x^2}\right)$	$-2 \leq x \leq -1$	0	$S = -(1+x)^2 + \frac{(1+x)^4}{2} + \dots + (-1)^n \frac{(1+x)^{2n}}{n}$
5	$y = \frac{e^x - e^{-x}}{2}$	$0,1 \leq x \leq 1$	0	$S = x + \frac{x^3}{3!} + \dots + \frac{x^{2n+1}}{(2n+1)!}$

Методичні вказівки

1. Алгоритм розв'язання задачі зводиться до циклу змінної x . У тілі циклу викликаються три функції – дві і комбінація бібліотечних (результат їх роботи вважається умовно точним значенням). Необхідно спроектувати та реалізувати користувацькі функції розрахунку $F(x) = S$ за двома вказаними вище алгоритмами.

2. Результати розрахунків надрукувати у вигляді таблиць:

Обчислення функції				
XУ	Y1	Y2	погр1	погр2
.....
.....
.....
.....

Тут X значення параметра; $Y1$ значення суми для заданого n ; $Y2$ значення суми для заданої точності; Y – точне значення функції; $\text{погр}1, \text{погр}2$ – відносні похибки наближених обчислень.

3. Для розрахунку чергового доданку суми необхідно (по можливості) використовувати значення вже обчисленого попереднього доданку.

Зміст звіту

1. Постановка задачі.
2. Варіант завдання.
3. Математична модель (формули, якими виконуються обчислення доданків ряду).
4. програма.
5. Отримані результати.

Приклад програми для лабораторної роботи

```
/**
Постановка задачі
"Обчислення функцій з використанням їх розкладання в степенний ряд"
Постановка задачі
Для x, що змінюється від a до b з кроком (b-a)/k,
де (k=10), обчислити функцію f(x),
використовуючи її розкладання в степенний ряд у трьох випадках:
а) для заданого n;
б) для заданої точності e(e=0.0001);
с) для "точного" значення (за бібліотечними функціями).
Для порівняння знайти відносну похибку обчислення функції значення функції
o_погр = ABS((точ_знач - набл_знач) / точ_знач)
Програма повинна містити три функції користувача,
виконують розрахунок значення Y трьома способами.
http://kit.znu.edu.ua/iLec/2sem/PROGR2/Lr2c/2/PM\_C++\_2.htm
y = (exp (x) - exp (-x)) / 2
0.1 <= x <= 1.0
k=20

an = x ** (2 * n + 1) / (2 * n + 1)!
a_нове=a_старе * x**2/( 2*(2n+3)*(n+1) )
n починається з НУЛЯ
**/

// додаємо до "початкового коду" програми ПРОТОТИПИ бібліотечних функцій
#include <iostream>
#include <stdio.h>
#include <math.h>
using namespace std; // використовуємо "стандартний простір імен" за
умовчанням
/* Прототипи функцій користувача */
double f1(double x); // розраховує значення f(x) через набір функцій
БІБЛІОЧКИ
double f2(double x, int n); // Розраховує значення f(x)
// через ряд Маклорена з фіксованою кількістю доданків n
double f3(double x, double eps); // Розраховує значення f(x)
// через ряд Маклорена із заданою точністю eps
// Головна програма
int main(int argc, char *argv[], char *env[]) { // "загальний вигляд"
заголовка ГОЛОВНОЇ процедури
setlocale(LC_ALL, "Російський"); // для "русифікації консолі"

cout << "Привіт із кодеблоку" << endl;
printf("Робота № 2 Варіант № 25\n");
double a=0.1L, b=1.0L; // інтервал аргументу x
int ktoc=10; // кількість точок значень функцій на відрізку [a, b]
int n=20; // кільк. доданків при розрахунку через ряд Маклорена з фіксованою
кількістю доданків
double x // значення аргумен
```


Результат

Привіт із кодеблоку
Робота №2 Варіант №25

x	y1	y2	y3	er1 %	er2 %
0,1	0,10017	0,10005	0,10017	0,12	2e-008
0,19	0,19115	0,19034	0,19115	0,42	9,3e-007
0,28	0,28367	0,2811	0,28367	0,91	9,4e-006
0,37	0,3785	0,37254	0,3785	1,6	5e-005
0,46	0,4764	0,46489	0,4764	2,4	5,3e-007
0,55	0,57815	0,55838	0,57815	3,4	2,2e-006
0,64	0,68459	0,65324	0,68459	4,6	7,3e-006
0,73	0,79659	0,7497	0,79659	5,9	2e-005
0,82	0,91503	0,84801	0,91503	7,3	5,1e-005
0,91	1,0409	0,94843	1,0409	8,9	8,6e-007
1	1,1752	1,0512	1,1752	11	2,1e-006

Process returned 0 (0x0) execution time : 0.050 s
Press any key to continue.

Робота № 2, мова програмування C "Функції та масиви"

Ціль: Організувати обробку масивів із використанням функцій, навчитися передавати масиви як параметри функцій.

Постановка задачі

Використовуючи функції, вирішити вказане у варіанті завдання. Масив повинен передаватися у функцію як параметр.

Варіанти

1. Визначити скільки елементів двовимірного масиву менше за будь-який елемент на головній діагоналі.
2. Написати функцію для обміну рядків двовимірного масиву за допомогою її відсортувати масив за елементами третього стовпця.
3. Написати процедуру для підсумовування матриць. З її допомогою скласти вихідну матрицю та транспоновану (тобто отриману поворотом вихідної на 90°).

4. Написати функцію для видалення рядка із двовимірного масиву. Рядки, що залишилися, повинні бути розташовані щільно, відсутні елементи замінюються 0. За допомогою розроблених функцій виключити з масиву рядки з номерами від A до B.
5. Визначити чи є матриця ортонормованою, тобто такою, що скалярний добуток кожної пари різних рядків дорівнює 0, а скалярний добуток рядка самої на себе дорівнює 1.
6. Елемент матриці є сідловою точкою, якщо він є найменшим у своєму рядку і найбільшим у своєму стовпці (або навпаки: найбільшим у своєму рядку і найменшим у своєму стовпці). Для заданої матриці визначити всі сідлові точки.
7. Написати процедуру обміну стовпця та рядки двовимірного масиву. З її допомогою поміняти місцями ті рядки та стовпці, перші елементи яких збігаються.
8. Написати функцію транспонування квадратної матриці (тобто повороту вихідної матриці на 90°). З її допомогою визначити, чи є задана матриця симетричною. (Матриця називається симетричною, якщо транспонована матриця дорівнює вихідній).
9. Написати функцію для обчислення суми елементів квадратної матриці, які розташовані нижче за головну діагональ. З її допомогою знайти максимальне значення такої суми у n матрицях.
10. Написати функцію, яка перевіряє чи є негативні елементи у зазначеному рядку двовимірного масиву. Видалити з масиву всі рядки з негативними елементами, віддалений рядок заповнюється 0 і переноситься на кінець масиву.
11. Написати функцію, яка перевіряє за зростанням або зменшенням упорядкований зазначений рядок двовимірного масиву. Упорядкувати за зростанням усі рядки двовимірного масиву, які неупорядковані за спаданням.
12. Написати функцію для пошуку максимального елемента у зазначеному рядку двовимірного масиву. Зрушити в двовимірному масиві всі рядки циклічно вправо на кількість елементів, що дорівнює максимальному елементу в цьому рядку.
13. Визначити чи можна у двовимірному масиві знайти такий стовпець, який розбиває масив на два так, що сума елементів у першому більша, ніж сума елементів у другому. Сам стовпець у частини, що розбиваються, не входить.
14. Обчислити добуток всіх стовпців масиву, у яких перший елемент більше елементів розташованих на головній та побічній діагоналі.
15. Заданий двовимірний масив. Знайти суму елементів першого шпальти без одного останнього елемента, суму елементів другого

шпальти без двох останніх, суму елементів третього шпальти без трьох останніх і т. д. Останній шпальт не обробляється. Серед знайдених сум знайти максимальну.

16. Заданий двовимірний масив $N \times N$. Дозволяється довільно переставляти елементи всередині будь-якого стовпця. Перевірити, чи можна виконавши кінцеву кількість перестановок у стовпцях, розташувати на побічній діагоналі елементи так, щоб він зростав.
17. Заданий двовимірний масив $N \times M$. Знайти в ньому підмасив 3×3 сума елементів якого максимальна. N та M можуть бути не кратні трьом.
18. Заданий двовимірний масив $N \times N$. Послідовно розглядаються квадратні підмасиви, верхній верхній елемент яких лежить на побічній діагоналі. У кожному підмасиві знаходиться максимальний елемент. Шляхом перестановок рядків і стовпців (цілком) елемент треба перемістити в правий верхній кут підмасиву. Перевірити чи вийшла на побічній діагоналі спадна послідовність елементів.
19. Задано рядок із N^2 цифр. Чи можна встановити, розбивши рядок на підрядки довжиною N , записати їх у рядки двовимірного масиву $N \times N$ по одній цифрі в одному елементі так, щоб вони в першому стовпці розташувалися в порядку зростання.
20. Знайти мінімальний із повторюваних елементів двовимірного масиву.
21. Знайти максимальний з елементів двовимірного масиву, що повторюються.
22. У двовимірному масиві знайти середнє арифметичне першого стовпця та кількість елементів у кожному з наступних стовпців, що перевищують середнє арифметичне попереднього стовпця.
23. Заданий одновимірний масив, що складається з N цілих чисел. Сформувати на його основі двовимірний масив $N \times N$ так, щоб сума елементів у першому стовпці дорівнювала першому елементу одновимірного масиву, сума елементів у другому стовпці дорівнювала другому елементу одновимірного масиву і т. д. Нулі не використовувати.
24. Визначити скільки елементів двовимірного масиву більше за мінімальний елемент на головній діагоналі.
25. Визначити, скільки елементів двовимірного масиву менше максимального елемента на головній діагоналі.

Алгоритм розв'язання багатьох задач зводиться до розробки необхідної за умовою задачі функції та її багаторазового застосування для ряду стовпців або рядків.

Рекомендується користуватися динамічними масивами.

Рекомендується також розробити функцію виведення на консоль значень елементів масиву і використовувати її для візуалізації значень до і після перетворення масивів.

Значення елементів масиву можна встановити з використанням функції отримання випадкових чисел – random чи rand.

Зміст звіту

1. Постановка задачі.
2. Варіант завдання
3. Текст програми.
4. Результат розв'язання конкретного варіанта.

Приклад рішення лабораторної роботи №2 (С)

```
/**
Постановка задачі
25. Визначити скільки елементів двовимірного масиву менше
максимального елемента на головній діагоналі.
**/

// додаємо до "початкового коду" програми ПРОТОТИПИ бібліотечних
функцій
#include <iostream>
#include <stdio.h>
#include <math.h>
#include <ctime>
#include <cstdlib>
using namespace std; // використовуємо "стандартний простір
імен" за умовчанням
// Головна програма
int main(int argc, char *argv[], char *env[]) { // "загальний
вигляд" заголовка ГОЛОВНОЇ процедури
setlocale(LC_ALL, "Російський"); // для "русифікації консолі"

cout << "Привіт із кодеблоку" << endl;
printf("Робота № 2-2 Варіант № 25\n");
printf("Визначити скільки елементів двовимірного масиву \nменше
максимального елемента на головній діагоналі\n");
/* визначимося з вихідними даними
використовуватимемо динамічний масив
розмірності nxm
За умовою завдання передбачається, що квадратна матриця n=m
Значення елементів "для налагодження" задаватимемо
```

```

набором випадкових цілих чисел
*/
int n, m; // Розмірність матриці
// int i, j; // поточні індекси
int maxd; // максимальний елемент головної діагоналі
int krez=0; // кількість елементів двовимірного масиву менша за
максимальний елемент на головній діагоналі
int * * mat = NULL; // покажчик на матрицю
printf("Введіть розмірність матриці (ціле число):");
scanf("%d", &n);
if (n < 0) {
printf("Помилка значення розмірності матриці...
n=%d\nДосвідання....\n", n);
return -1;
}
m=n;

// "Створимо матрицю" виділимо під неї ВП
mat = (int * *) calloc (sizeof (int *), n); // покажчик на
вектор покажчиків int *
if (mat == NULL) {
printf("*** Помічено до об'єднаної пам'яті для mat => %f Kb\n",
(float) (sizeof (int *) * n) / 1024.0);
return -1;
};
for(int ii=0; ii < n ; ii++) mat[ii]=NULL; // Вектор
ініціалізували NULL - це не обов'язково
for(int ii=0; ii < n ; ii++) { // у кожен елемент вектора
запишемо адресу "рядки" матриці
mat[ii]=( int *) calloc( sizeof(int), m );
if (mat[ii] == NULL) {
printf("*** Використовується для збереження пам'яті для mat[%d]
=> %f Kb\n",
ii, (float) (sizeof(int) * m)/1024.0);
// Необхідно раніше захоплену ВП повернути системі.
for(int iii=0;iii <= ii; iii++) if (mat[iii] !=NULL)
free(mat[iii]);
if (mat! = NULL) free (mat);
return -1;
};
};
// пам'ять під матрицю виділено задамо значення елементів
// Ініціалізація датчика випадкових чисел.
//Включення автоматичної рандомізації
srand(time(0));

for(int i=0; i < n; i++)
for(int j=0; j < m; j++)
mat[i][j]=rand() % 100; // rand() повертає випадкове число з
[0,RAND_MAX],
// Зазвичай RAND_MAX=32767, операція % - залишок від розподілу
// Виведемо вихідну матрицю

```

```

printf("початкова матриця\n");
for(int i=0; i < n; i++) {
for(int j=0; j < m; j++)
printf("%4d", mat[i][j]);
printf("\n");
};
// знайдемо maxd максимальний елемент головної діагоналі
maxd=mat[0][0];
for(int i=1; i < n; i++)
if (mat[i][i] > maxd) maxd=mat[i][i];
printf("максимальний елемент головної діагоналі %d\n", maxd);

// krez знайдемо кількість елементів двовимірного масиву менше
максимального елемента на головній діагоналі
// Елементи, що лежать на головній діагоналі враховувати не
будемо
for(int i=0; i < n; i++)
for(int j=0; j < m; j++){
if ( i == j ) continue;
if (mat [i] [j] <maxd) krez ++;
};
printf("необхідна кількість елементів %d\n", krez);
// Повернімо ВП системі
if (mat != NULL ) {
for(int ii=0; ii < n; ii++)
if (mat[ii] != NULL) free(mat[ii]);
free(mat);
};
return 0;
}

```

Результат роботи

```

Привіт із кодеблоку
Робота №2-2 Варіант №25
Визначити скільки елементів двовимірного масиву
менше максимального елемента на головній діагоналі
Введіть розмірність матриці (ціле число):5
вихідна матриця
80 17 89 41 2
65 89 56 65 3
67 77 0 21 42
8 67 72 74 9
95 65 27 47 13
максимальний елемент головної діагоналі 89
кількість елементів 18

Process returned 0 (0x0) execution time : 2.670 s
Press any key to continue.

```