**ЛАБОРАТОРНА РОБОТА № 1**

**Тема роботи:** Створення простих сценаріїв на мові PHP.

**Мета роботи:** Одержання практичних навиків розробки простих сценаріїв з використанням PHP.

**1. Теоретичні відомості**

РНР (створений у якості передобробника гіпертекстів — Hypertext Preprocessor) увібрав в себе всю простоту сценаріїв і міць засобів розробки серверної частини Web-додатків. Якщо взяти до уваги його простий C-подібний синтаксис, підтримку найрізноманітніших баз даних, об’єктно-орієнтовного підходу і можливість безкоштовного використання, то стане зрозуміла зростаюча популярність мови РНР.

### *Принципи обробки сценаріїв PHP*

Коли від клієнтського броузера надходить запит на одержання звичайної сторінки HTML, то Web-сервер виконує наступну просту послідовність дій:

1. Аналіз HTTP-запиту.

2. Пошук потрібного серверного ресурсу (сторінки).

3. Передача даних назад клієнту.

У випадку запиту сторінки, до складу якої входять PHP-сценарії, наведена вище послідовність дій ускладнюється. Web-сервер в цьому випадку повинен виконати аналіз файлу, до якого відбувається запит, а при виявленні операторів PHP викликати відповідний інтерпретатор. У трохи спрощеному вигляді послідовність дій, яка виконується, виглядає в такий спосіб.

1. Аналіз HTTP-запиту.

2. Пошук необхідного серверного ресурсу (сторінки).

3. Аналіз сценаріїв, виявлених на сторінці, шляхом виклику відповідного інтерпретатора.

4. Передача назад клієнтові результуючої сторінки.

Необхідно відзначити, що на сторінці, що передається клієнтському броузеру, вже не міститься ніяких сценаріїв. Їх місце зайняли дані, згенеровані в процесі їхньої інтерпретації. В загальному випадку такими даними може бути не тільки HTML-код, але й різні фрагменти на мовах клієнтських сценаріїв (JavaScript і т.д.).

### *Включення фрагментів РНР у HTML-код*

Фрагмент мовою РНР можна включити в текст Web-сторінки декількома способами: за допомогою стандартних дескрипторів; за допомогою коротких дескрипторів; за допомогою дескрипторів script; за допомогою дескрипторів в стилі ASP.

#### *Стандартні дескриптори*

Для включення в сторінки HTML коду РНР найчастіше використовуються стандартні дескриптори. Це пояснюється їхньою наочністю і зручністю у використанні. Наприклад,

<?php

print “This is a simple test!”;

?>

#### *Короткі дескриптори*

Короткі дескриптори забезпечують найбільш компактний перехід до коду РНР.

<? echo “This is a simple test!”; ?>

За замовченням короткі дескриптори не використовуються, їх потрібно спеціально активізувати. Це можна зробити двома способами.

вказати ключ -enable-short-tags при компіляції модуля РНР;

додати параметр short\_open\_tag у файл php.ini.

#### *Дескриптори script*

Деякі текстові редактори помилково приймають код РНР за код HTML (тобто візуально відображуваний текст), що порушує правильну інтерпретацію Web-сторінки. Ця проблема вирішується шляхом використання дескрипторів script.

<script language=“php”>

<?php print “This is a simple test!”; ?>

</script>

#### *Дескриптори в стилі ASP*

Четвертий і останній спосіб включення коду мовою РНР в текст HTML-сторінки полягає у використанні дескрипторів в стилі ASP. Вони нагадують короткі дескриптори, однак замість знаку питання використовується символ %.

<%php echo “This is a simple test!”; %>

### *Змінні*

В мові РНР імена всіх змінних починаються із символу долара ($) і наступного за ним імені змінної. Імена змінних чуттєві до регістра. Тобто імена $username і $UserName є іменами двох різних змінних. Відповідно до правил іменування необхідно, щоб ім'я змінних починалося з букви або символу підкреслення, за яким може йти будь-яка кількість буквено-цифрових символів або символів підкреслення. Наприклад,

<?php

$username = ‘Barry’;

$UserName = ‘White’;

echo “$username $UserName”;

?>

Прикладами неправильних імен змінних є наступні: *$4name*, *$&Test*, *$8+abc*.

### *Типи даних*

Як і у будь-якій іншій мові, в мові РНР існують різні вбудовані типи даних.

* цілий
* тип з плаваючою крапкою
* логічний (можуть приймати одне з двох значень — істина (true) або хиба (false))
* строковий
* масиви
* класи
* тип NULL

Однак на відміну від строго типізованих мов, наприклад С++, у РНР тип змінної не описується, а визначається виходячи з контексту її використання.

$var = 2; // $var - ціле число 2

$var = “2”; // $var – рядок, який містить 2

$var = array(2); // $var - це масив, що містить // один елемент - 2

#### Строковий тип

Рядок у PHP — це послідовність байтів. У PHP немає вбудованої підтримки формату Unicode. Строкові константи (літерали) можна визначити за допомогою одинарних або подвійних лапок. Спеціальні символи (наприклад, \’ і \\) у строкових літералах, обмежених одинарними лапками, не розпізнаються. Обробка строкових констант у подвійних лапках відбувається зовсім по іншому. Зокрема, вбудовані змінні будуть коректно розпізнані інтерпретатором РНР (тобто інтерпольовані) і замість них будуть підставлені їхні поточні значення.

Наприклад, рядок

‘The age is: $age’

буде виведений у вихідний потік без змін. Однак якщо припустити, що поточним значенням змінної $age є 12, то при виводі рядок

“The age is: $age”

буде перетворено до наступного виду

The age is: 12

У мові PHP визначена єдина операція конкатенації рядків з використанням операції . (крапка). Рядок можна розглядати як масив символів, нумерація яких починається з нуля, і звертатися до окремого символу, вказуючи у фігурних дужках його номер. Наприклад, якщо в змінній $str міститься рядок “abcdefgh”, то при обробці наступних двох рядків

$first = $str{0}; // ‘а'

$last = $str{strlen($str)-1}; // ‘h’

отримаємо такий результат

a

h

### *Операції*

Операції використовуються для маніпулювання над значеннями однієї або декількох змінних. Найпростішим прикладом може послужити операція додавання в арифметиці. Операції мови PHP в основному аналогічні операціям у мові С++.

Існує декілька груп операцій.

* Арифметичні, які визначають найпростіші математичні дії над змінними.
* Операції присвоювання, призначені для зміни значення деякої змінної.
* Логічні, що дозволяють обчислювати булеві вирази.
* Порозрядне, призначені для роботи з двійковими представленнями чисел.

Це ті операції, які можна зустріти практично в кожній мові програмування. В той же час в мові PHP визначено досить специфічні операції.

* Конкатенація (звичайно ця операція реалізується за допомогою бібліотечної функції).
* Обробки помилки.
* Умовна операція.

Нижче в таблиці 1 наведено перелік основних операцій мови PHP в порядку їхнього пріоритету.

### 

### *Керуючі оператори*

Кожен сценарій PHP складається з виразів. До складу виразу може входити операція присвоювання, виклик функції, а також керуючі оператори (умовні оператори, цикли, блоки розгалуження й інші). Як правило, кожен вираз закінчується крапкою з комою (;). Вирази можуть бути згруповані й утворювати складний вираз.

Таблиця 1

Основні операції мови РНР

|  |  |
| --- | --- |
| Операція | Опис |
| () | Визначення пріоритету |
| [] | Доступ до елемента масиву за індексом |
| ! | Логічне заперечення |
| ~ | Порозрядне заперечення |
| ++,-- | Інкремент, декремент |
| @ | Обробка помилки |
| \* / % | Множення, ділення, ділення із залишком |
| + - . | Додавання, віднімання, конкатенація |
| << >> | Порозрядне ліве зміщення, праве зміщення |
| < <= > >= | Менше, менше дорівнює, більше, більше дорівнює |
| == != === !== | Дорівнює, не дорівнює |
| & | Порозрядне І |
| ^ | Порозрядне виключаюче АБО (XOR) |
| | | Порозрядне АБО |
| && | Логічне І |
| || | Логічне АБО |
| ?: | Умовна операція |
| = += -= \*= /= .=  %= &= != ~= <<= >>= | Присвоювання |
| And | Логічне І |
| Xor | Логічне виключає АБО |
| Or | Логічне АБО |

#### *if — else — elseif*

Дана конструкція є однією з найважливіших практично в будь-якій мові програмування. І PHP не є виключенням. Цей керуючий оператор дозволяє виконувати або пропускати визначені фрагменти коду при заданих умовах. Загальний вид оператора if наведений нижче.

if (умова 1) {

блок коду 1

} elseif (умова 2) {

...

} elseif (умова N) {

блок коду N

} else {

альтернативний блок коду

}

Коли умова 1 істина (тобто її результатом є значення true), виконується лише блок коду 1. В противному випадку перевіряється умова 2 і т.д. Якщо всі умовні вирази виявилися хибними, виконується альтернативний блок коду, визначений оператором else. Наприклад,

if ($a > $b) {

print “a is bigger than b”;

} elseif ($a == $b) {

print “a is equal b”;

} else {

print “b is bigger than a”;

}

#### *while*

Цикл найпростіше реалізувати за допомогою наступної конструкції.

while (умова)

{

блок коду;

}

Блок коду багаторазово виконується доти, поки умова залишається істиною. Умова перевіряється щоразу на початку циклу. Якщо умова приймає значення false відразу на першій ітерації, то оператори тіла циклу не виконаються жодного разу. Нижче приведений приклад використання оператора while для виводу чисел від 1 до 10.

<?php

$i = 1;

while ($i <= 10) {

echo $i++;

}

?>

#### *do — while*

Цей цикл дуже схожий не попередній, за винятком того, що умова перевіряється наприкінці кожної ітерації після виконання операторів тіла циклу. Іншими словами гарантується, що оператори тіла циклу будуть виконані хоча б один раз.

do

{

блок коду;

}

while (умова);

#### *for*

У мові PHP найбільш складні цикли можна реалізувати з використанням оператора for. Для цього використовується наступний синтаксис.

for (вираз 1; вираз 2; вираз 3)

{

блок коду;

}

Вираз 1 виконується один раз на початку першої ітерації без перевірки деяких умов. Як правило, цей вираз використовується для ініціалізації змінної-лічильника циклу. На початку кожної ітерації перевіряється вираз 2. Якщо воно істина, то виконується блок коду, у противному випадку виконання циклу припиняється. Наприкінці кожної ітерації виконується вираз 3. Кожен з виразів може бути порожнім.

#### *foreach*

Цей оператор з'явився в РНР версії 4. Він дозволяє виконувати задані дії над всіма елементами масиву. Масиви більш докладно будуть розглянуті нижче. При використанні оператора foreach використовується наступний синтаксис.

foreach (масив as $value) {

}

foreach (масив as $key=>$value) {

}

У першому випадку на кожній ітерації значення поточного елемента масиву привласнюється змінній $value і поточний показник масиву переміщується на один елемент уперед (тобто на наступній ітерації буде привласнене значення наступного елемента). В другому випадку виконується те ж саме, за винятком того, що значення ключа масиву привласнюється змінній $key. Наприклад,

<?php

$arr = array(‘one’, ‘two’, three’);

foreach ($arr as $value) {

echo ‘Value: ’.$value.’<br>’;

}

?>

#### *switch*

Даний оператор аналогічний декільком інструкціям if. Найчастіше значення однієї і тієї ж змінної необхідно порівняти з декількома іншими значеннями й в залежності від цього виконати різні дії. Тоді найкраще скористатися саме оператором switch. Наприклад,

<?php

switch ($value) {

case ‘apple’ :

echo ‘This is an apple!‘;

break;

case ‘pear’ :

echo ‘This is a pear!‘;

break;

default :

echo ‘Hm.. Who knows...’;

}

?>

#### *Оператори break і continue*

Оператор break дозволяє зупинити виконання циклів і інших керуючих конструкцій, реалізованих за допомогою операторів for, foreach, while, do…while,switch...

Оператор continue можна застосовувати всередині циклів для зупинки виконання частини, що залишилася, поточної ітерації і переходу до наступної ітерації.

### *Масиви*

В мові РНР масив можна створити двома способами. Для створення скалярних змінних використовується операція присвоювання. Її можна застосовувати і для створення масивів. Присвоювання значення елементу масиву, що до цього ще не існував, приведе до створення цього масиву. Наприклад, припустимо, що в даний момент масив $mass ще не існує. Тоді наступний вираз приведе до його створення.

$mass[0] = 5;

Другий спосіб створення масиву полягає у використанні оператора array. Параметри цього оператора задають значення, що будуть розміщені в новому масиві, а при необхідності і відповідні ключі. Якщо масив планується використовувати традиційним способом (тобто без ключів), то в операторі array можна задати тільки значення. (При цьому інтерпретатор РНР автоматично привласнить ключам цілі значення.) Наприклад,

$mass = array(10, 20, 30, 40);

При такому присвоюванні буде створений звичайний масив з чотирма елементами і ключами 0, 1, 2 і 3. Якщо потрібно вказати інші ключі, то це можна здійснити в такий спосіб.

$mass = array(1 => 10, 2 => 20, 3 => 30, 4 => 40);

В наступному рядку створюється масив, що буде відповідати хеш-таблиці Perl.

$ages = array(“Ivan” => 40, “Mary” => 17, “Alex” => 25);

До окремих елементів масиву доступ можна одержати за індексом, як і в інших традиційних мовах програмування. При цьому взятий у квадратні дужки індекс є ключем шуканого значення. Дужки використовуються незалежно від того, цілочислений або строковий ключ використовується. Наприклад, значення елемента масиву $ages із ключем “Mary” можна одержати в такий спосіб.

print(“Mary is $ages[‘Mary’] years old <br />”);

**2. Завдання для виконання**

2.1. Перед виконанням роботи ознайомитись з теоретичними відомостями.

2.2. Проаналізувати умову задачі.

2.3. Виконати завдання 1-6 із врахуванням індивідуальних варіантів завдань. Завдання 1-2 є необхідним розв’язком при отримані позитивної оцінки 60-74 бали. Завдання 1-4 - для отримання оцінки 75-89. Завдання 1-6, для отримання відмінної оцінки: 90-100 балів.

2.4. Результати роботи оформити у формі звіту, який доцільно оформити у деталізованому описі послідовності виконання роботи, доповнюючи екранними копіями та лістингом відповідних скриптів.

2.5. При захисті лабораторної роботи представити функціональну веб-сторінку, де розмістити особисту інформацію (ПІП, Група, Варіант) з можливістю навігаційних переходів між окремими завданнями до лабораторної роботи та їх розв’язками.

2.6. Підготувати відповіді на контрольні питання.

2.7. Сформувати відповідні висновки.

**3. Зміст звіту**

3.1. Тема та мета роботи.

3.2. Коротко основні теоретичні відомості.

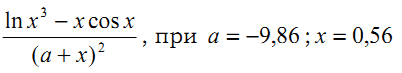
3.3. Відобразити отримані результати (п. 2.2-2.4) у вигляді покрокового опису послідовності виконання завдань, які виносилися на лабораторну роботу.

3.4. Надати пропозиції по розширенню функціональності розроблених сценаріїв.

3.5. Висновки за результатам виконаної роботи.

**4. Приклади розв’язання типових завдань**

Завдання №1. Обчислити значення виразу.



## Лістинг скріпта «lab\_1\_t1.php»

## *<?php*

## *echo '<html><body>';*

## *echo 'Лабораторна робота №1<br>';*

## *echo 'Варіант 2 <br>';*

## *echo 'Обчислення значення виразу <br>';*

## *$a=-9.86;*

## *$x=0.56;*

## *$y=( log( pow($x,3) )-$x\*cos($x) ) /pow($a+$x,2);*

## *echo "Отримані результати a = $a x = $x y = $y<br>";*

## *echo '</html>';*

## *?>*

Завдання №2. Дано натуральне число. Скільки разів перша цифра, починаючи зліва, зустрічається в даному числі.

## Лістинг скріпта «lab\_1\_t2.php»

## *<html>*

## *<head>*

## *<title>Лабораторна робота 1</title>*

## *</head>*

## *<body>*

## *<h2>Лабораторна робота №1</h2>*

## *<h3>Завдання 2</h3>*

## *<?php*

## *$n=1211324; // наше початкове число*

## *$t=$n; // шукаємо першу чифру числа*

## *while ($t>0) {*

## *$r=$t%10; $t=floor($t/10);}*

## *$t=$n; // порівнюємо першу цифру числа з іншии*

## *while ($t>0){*

## *$k=$t%10; $t=floor($t/10);*

## *if ($k==$r) {$a++;} }*

## *echo("Цифра $r зустрічається в числі $n S $a раз<br>" );*

## *?>*

## *</body>*

## *</html>*

Завдання №3. Скласти програму, яка з’ясовує, скільки різних елементів в заданому одновимірному масиві.

## Лістинг скріпта «lab\_1\_t3.php»

## *<html>*

## *<head>*

## *<title>Лабораторна робота 1</title>*

## *<meta httpSequiv="contentStype" content="text/html; charset=windowsS1251">*

## *</head>*

## *<body>*

## *<h2>Лабораторна робота №1</h2>*

## *<h3>Завдання 3</h3>*

## *<?php*

## *$m=array(12,3,2,4,2,1,S5,3,5,23,3,4,2,1); // наш масив*

## *sort($m); // Відсортуємо масив за зростанням*

## *$k=1; // кількість різних елементів*

## *for($i=0;$i<count($m)S1;$i++) {*

## *if ($m[$i]!=$m[$i+1]) $k++;};*

## *echo("В масиві<br>" );*

## *print\_r($m);*

## *echo("<br>Кількість різниц цифр рівне $k<br>" );*

## *?>*

## *</body>*

## *</html>*

Завдання №4. В одновимірному числовому масиві всі додатні числа замінити нулями.

## Лістинг скріпта «lab\_1\_t4.php»

## *<?php*

## *echo("<head><title>Лабораторна робота 1</title></head>*

## *<h2>Приклад</h2>*

## *Заміна додатніх елементів массиву на нульові значення.<br>");*

## *$mas=array(23,12,34,S5,23,3,4,S5,S6);*

## *echo("Задано масив<br>");*

## *print\_r($mas);*

## *echo('<br>Отримали наступний<br>');*

## *foreach($mas as $key => $vol)*

## *{if ($vol>0) $mas[$key]=0;}*

## *print\_r($mas);*

## *?>*

Завдання №5. Дано ціле число визначити чи є воно паліндромомом.

Для перевірки цього факту створимо число, яке записано тими ж цифрами але в зворотному напрямку. Як приклад, з числа 6753 утворимо число 3576. Після створення «зворотного» числа порівняємо його з наданим. Оформимо цю операцію за допомогою функції користувача. Для створення «зворотного» числа будемо «відрізати» по одній літері і доклеювати її до нового.

## Лістинг скріпта «lab\_1\_t5.php»

## *<?php*

## *function palind($x)*

## *{*

## *$t=$x; $p=0;*

## *while ($t > 0) {*

## *$p=$p\*10 + $t%10;*

## *$t=($t]$t%10)/10;*

## *}*

## *return $p==$x;*

## *}*

## *printf("%d", palind(121));*

## *?>*

Завдання №6. Переписати зміст текстового файлу test.txt в файл test\_good.txt, але без порожніх рядків.

## Лістинг скріпта «lab\_1\_t6.php»

## *<?php*

## *// Зчитаємо текстовий файл в масив lines $lines=file('test.txt); // Видалимо в масиві "порожні" рядки // враховуючи ознаки кінця рядка та перехід на // наступний рядок*

## *for($i=0; $i<count($lines);$i++) { if (strlen($lines[$ij) == 2)*

## *{unset($lines[$i]); // видалення елементу масива*

## *echo "Видаляється рядок з номером ".$i."<br>";J*

## *}*

## *// Записуємо масив lines до файлу*

## *$fp=fopen( "test\_good.txt","w");*

## *// implode() перетворює масив до рядка*

## *fputs($fp,implode("",$lines));*

## *fclose($fp);*

## *echo "--кінець--";*

## *?>*

**5. Контрольні запитання**

* 1. Як відбувається обробка запиту до сторінки, яка включає сценарій РНР?
  2. Назвіть вбудовані типи даних РНР.
  3. Назвіть основні керуючі оператори мови РНР. Порівняєте їхній синтаксис з написанням аналогічних операторів С++.
  4. У чому особливість масивів у РНР?
  5. Охарактеризуйте основні функції РНР для роботи з масивами.
  6. Назвіть основні способи звертання до елементів масиву в РНР.
  7. Що таке функція?
  8. Дайте визначення формальних і фактичних параметрів функції.

**6. Індивідуальні завдання**

|  |  |  |
| --- | --- | --- |
|  | Завдання №1 | Завдання №2 |
| № варіанту | Обчислити значення виразу. | Скласти програму і за її допомогою обчислити для декількох значень |
| 1,16 |  | Знайти добуток цифр даного натурального числа. |
| 2,17 |  | Визначити, чи є задане натуральне число паліндромом, тобто таким, що його десятковий запис читається однаково зліва на право і справа наліво. |
| 3,18 |  | Задано натуральне число n. Скласти програму, яка підраховує кількість цифр в числі n, суму цифр числа n та з’ясовує, чи входить цифра ’3’ до запису числа n. |
| 4,19 |  | Задано натуральне число n. Скласти програму, яка знаходить суму m перших цифр числа n та суму m останніх цифр числа n. |
| 5,20 |  | Задано натуральне число n. Скласти програму, яка змінює порядок цифр числа n на обернений. |
| 6,21 |  | Задано натуральне число n. Скласти програму, яка переставляє першу та останню цифру цього числа. |
| 7,22 |  | Знайти добуток цифр даного натурального числа. |
| 8,23 |  | Визначити, чи є задане натуральне число паліндромом, тобто таким, що його десятковий запис читається однаково зліва на право і справа наліво. |
| 9,24 |  | Задано натуральне число n. Скласти програму, яка підраховує кількість цифр в числі n, суму цифр числа n та з’ясовує, чи входить цифра ’3’ до запису числа n. |
| 10,25 |  | Задано натуральне число n. Скласти програму, яка знаходить суму m перших цифр числа n та суму m останніх цифр числа n. |
| 11,26 |  | Задано натуральне число n. Скласти програму, яка змінює порядок цифр числа n на обернений. |
| 12,27 |  | Задано натуральне число n. Скласти програму, яка переставляє першу та останню цифру цього числа. |
| 13,28 |  | Визначити, чи є задане натуральне число паліндромом, тобто таким, що його десятковий запис читається однаково зліва на право і справа наліво. |
| 14,29 |  | Задано натуральне число n. Скласти програму, яка підраховує кількість цифр в числі n, суму цифр числа n та з’ясовує, чи входить цифра ’3’ до запису числа n. |
| 15,30 |  | Задано натуральне число n. Скласти програму, яка знаходить суму m перших цифр числа n та суму m останніх цифр числа n. |
|  | Завдання №3 | Завдання №4 |
| № варіанту | Скласти програму і за її допомогою обчислити для декількох значень | Скласти програму і за її допомогою обчислити для декількох значень |
| 1,16 | Знайти найменший елемент в одновимірному числовому масиві. | Вивести на друк максимальний елемент масиву A(5,5) та його індекси. |
| 2,17 | Знайти найбільший елемент в одновимірному числовому масиві. | Вивести на друк мінімальний елемент масиву FR(4,6) та його індекси. |
| 3,18 | Перевірити, чи є в одновимірному числовому масиві від’ємні числа. Якщо є, то надрукувати їх індекси. | Максимальний елемент масиву Q(10,2) замінити на 10. |
| 4,19 | В одновимірному числовому масиві всі додатні числа замінити нулями. | Всі від’ємні елементи масиву MN(8,4) замінити їх квадратами. |
| 5,20 | Замінити в одновимірному числовому масиві числа, які більше 5, на число 0,2. | Підрахувати кількість нульових елементів двовимірного масиву R(9,4). |
| 6,21 | Визначити суму невід’ємних елементів в одновимірному числовому масиві. | Знайти суму всіх елементів матриці DR(15,3). |
| 7,22 | Визначити число від’ємних і невід’ємних елементів в одновимірному числовому масиві. | Вивести на екран суму елементів рядків двовимірного масиву MI(9,5). |
| 8,23 | В одновимірному числовому масиві замінити від’ємні числа їх квадратами. | Надрукувати суму елементів рядків двовимірного масиву MI(7,4). |
| 9,24 | Визначити, чи є одновимірний числовий масив упорядкованим за спаданням. | Надрукувати максимальні елементи стовпчиків матриці AC(5,5). |
| 10,25 | Знайти номер максимального за модулем елемента в одновимірному числовому масиві. | Надрукувати максимальні елементи рядків матриці F(4,6). |
| 11,26 | Знайти середнє арифметичне чисел одновимірного масиву. | Обчислити середнє арифметичне елементів масиву L(2,20). |
| 12,27 | Задано числовий масив a1, a2,..., a10. Збудувати масив b1, b2,..., b10, в якому b1= a1, b2= a1+ a2, b3=a1+a2+ +a3 і т.д. | Сформувати квадратну матрицю порядку 6, в якій елементи головної діагоналі дорівнюють 1, а всі інші – 10. |
| 13,28 | Перевірити, чи є в одновимірному числовому масиві хоча б одна пара протилежних чисел. | Задано прямокутну матрицю E(5,4). Отримати нову матрицю, поділивши всі невід’ємні елементи на 5, а всі  від’ємні елементи — на 10. |
| 14,29 | Знайти індекси всіх ненульових елементів в одновимірному числовому масиві. | Визначити номери рядків прямокутної матриці, які складаються тільки з додатних чисел. |
| 15,30 | Перевірити, чи є в одновимірному числовому масиві хоча б одна пара взаємно обернених чисел. | Визначити номери рядків прямокутної матриці, які складаються тільки із від’ємних елементів. |

|  |  |  |
| --- | --- | --- |
|  | Завдання №5 | Завдання №6 |
| № варіанту | Скласти програму для обчисленьвикористовуючи процедури та функції. | Скласти програму для обчислень використовуючи процедури та функції. |
| 1,16 | Скласти програму підрахунку загальної кількості додатних елементів в масивах A(8) та B(8). | Із рядка вилучити середню букву, якщо довжина рядка непарна, інакше - вилучити дві середні букви. |
| 2,17 | Знайти суму додатних елементів одновимірних масивів E(7) та F(7) | Дано два рядки. Надрукувати результат рівності перших *n* символів. |
| 3,18 | Знайти різницю між максимальними елементами одновимірних масивів A(9) та G(9). | Дано послідовність символів. Надрукувати всі слова, відмінні від слова hello. |
| 4,19 | Знайти різницю мінімальних елементів масивів A(7) та B(7). | Дано послідовність слів. Надрукувати всі слова послідовності, що зустрічаються у ній один раз. |
| 5,20 | Скласти програму знаходження максимального з трьох чисел a,b,c знаходженням максимального з двох чисел. | Дано послідовність слів. Надрукувати всі слова, попередньо вилучивпіи зі слова всі включения останньої букви крім неї самої. |
| 6,21 | Знайти частку K=L/M, де L – кількість нульових елементів масиву B(8), а M – кількість нульових елементів масиву C(8). | Дано текст з маленьких латинських букв, за яким йде крапка. Надрукувати букви, що входять в текст не менше двох раз. |
| 7,22 | Знайти добуток мінімальних елементів масивів A(9) та B(9). | Дано текст з маленьких латинських букв, за яким йде крапка. Надрукувати букви, що входять в текст не менше трьох раз. |
| 8,23 | Знайти добуток максимальних елементів масивів P(8) та Q(8). | Дано текст з маленьких латинських букв, за яким йде крапка. Надрукувати букви, що входять в текст по одному разу. |
| 9,24 | Знайти різницю R=M-L, де M – сумадодатних елементів масиву A(7), а L – сума додатних елементів масиву B(7). | Дано текст з прописними латинськими літерами, за яким йде крапка. Надрукувати літери, що йдуть в тексті до літери "f. |
| 10,25 | Знайти значення виразу (3E+5F)/10, де E – мінімальний елемент масиву A(9), а F – мінімальний елемент масиву B(9). | Дано текст з прописними латинськими літерами, за яким йде крапка. Надрукувати літери, що йдуть в тексті після літери "f. |
| 11,26 | Задано два масиви A(5) та B(5). Змінній s присвоїти значення –1, якщо максимальний елемент масиву A більше максимального елемента масиву B, 0 та 1 відповідно при рівності та меншості. | Дано текст з прописними латинськими літерами, за яким йде крапка. Надрукувати літери, що йдуть в тексті до літер “rtf”. |
| 12,27 | Задано два масиву X(5) та Y(5). Змінній F присвоїти значення 1, якщо сума квадратів елементів масиву X більше суми квадратів елементів масиву Y, і –1 в протилежномувипадку. | Дано послідовність слів. Надрукувати ті слова послідовності, що відмінні від останнього слова і задовольняють таку властивість - слово складається тільки з голосних літер. |
| 13,28 | Обчислити значення змінної s=s1+s2, де s1 – сума елементів масиву A1(7), абсолютна величина яких не перевищує числа 7, а s2 – сума елементів масиву A2(7), що володіють тією ж властивістю. | Дано текст з прописними латинськими літерами, за яким йде крапка. Надрукувати літери, що входять в текст чотири рази. |
| 14,29 | Обчислити значення змінної M=max(M1,M2), де M1 – мінімальний елемент масиву B1(8), а M2 –мінімальний елемент масиву B2(8). | Дано текст з маленьких латинських букв, за яким йде крапка. Надрукувати букви, що ідуть в тексті поруч з буквою 'z’. |
| 15,30 | Задано два масиви X(7) та Y(7), обчислити значення змінної R=min(K1,K2), де K1 – кількістьневід’ємних елементів масиву X, а K2 – кількість невід’ємних елементів масиву Y. | Дано послідовність слів. Надрукувати ті слова послідовності, що відмінні від останнього слова і задовольняють таку властивість - слово збігається з початковим відрізком латинського алфавіту. |