**, Pętle, tablice –**

Dział A (10pkt)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Numery z dziennika** | | | | | | | | | | | | | | | |
| **NRD** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
|  | **Zadania do realizacji - nr** | | | | | | | | | | | | | | | |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|  | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 1 |
|  | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 1 | 2 | 3 | 4 | 7 |
|  | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|  | 13 | 14 | 15 | 16 | 17 | 18 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | 16 | 17 | 18 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

W skryptach dane podajemy poprzez formularz z projektu C

Utwórz następujące skrypty:

1. Wyświetli na ekranie 10 kolejnych liczb całkowitych począwszy od **n** podanego z formularza.
2. Wyświetli na ekranie 10 kolejnych liczb całkowitych począwszy od **n** podanego z formularza w kolejności malejącej.
3. Obliczy i wyświetli na ekranie sumę kolejnych 10 liczb całkowitych począwszy od **n** podanego z formularza.
4. Wyświetli na ekranie 10 kolejnych parzystych liczb całkowitych począwszy od **n** podanego z formularza
5. Obliczy sumę elementów ciągu arytmetycznego, składającego się ze 100 elementów, z których każdy kolejny jest większy od poprzedniego o 10. Pierwszy element wynosi **a**.
6. Obliczy ilość cegieł w piramidzie o podstawie 10 x 10 cegieł (następna warstwa 9x9 itd.).
7. Obliczy ilość cegieł w ścianie o podstawie X cegieł, wysokości Y rzędów cegieł, gdzie każdy kolejny stawiany rząd cegieł jest o Z cegieł krótszy od poprzedniego (niższego).
8. Oblicza wagę ściany z 8. podpunktu, przy założeniu, że jedna cegła waży K kilogramów.
9. Symulujący działanie prostego kalkulatora ( + i -).
10. Symulujący działanie prostego kalkulatora ( \* i /).
11. Oblicza sumę dwóch liczb podanych przez użytkownika. Algorytm ma działać tak długo, aż suma podanych liczb będzie wynosiła 0.
12. Algorytm oblicza silnię wprowadzonej przez użytkownika liczby.
13. Wypisuje na ekran ciąg **n** podanego z formularza. liczb, z których każda kolejna jest kwadratem poprzedniej liczby z ciągu. Wartość pierwszej liczby podaje użytkownik.
14. Oblicza średnią z **n (** podanego z formularza.) losowych liczb z przedziału <50,100>.
15. Oblicza sumę z **n (** podanego z formularza.) losowych liczb z przedziału <50,100>.
16. Napisz skrypt tworzący tabliczkę mnożenia dla dowolnej $n np: $n=7

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|  |  |  |  |  |  |  |
| **2** | 4 | 6 | 8 | 10 | 12 | 14 |
|  |  |  |  |  |  |  |
| **3** | 6 | 9 | 12 | 15 | 18 | 21 |
|  |  |  |  |  |  |  |
| **4** | 8 | 12 | 16 | 20 | 24 | 28 |
|  |  |  |  |  |  |  |
| **5** | 10 | 15 | 20 | 25 | 30 | 35 |
|  |  |  |  |  |  |  |
| **6** | 12 | 18 | 24 | 30 | 36 | 42 |
|  |  |  |  |  |  |  |
| **7** | 14 | 21 | 28 | 35 | 42 | 49 |

1. Napisz skrypt tworzący tabliczkę mnożenia dla dowolnej $n. np: $n=7

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  |  |  |  |  |  |  |
| 2 | 4 | 6 | 8 | 10 | 12 | 14 |
|  |  |  |  |  |  |  |
| 3 | 6 | 9 | 12 | 15 | 18 | 21 |
|  |  |  |  |  |  |  |
| 4 | 8 | 12 | 16 | 20 | 24 | 28 |
|  |  |  |  |  |  |  |
| 5 | 10 | 15 | 20 | 25 | 30 | 35 |
|  |  |  |  |  |  |  |
| 6 | 12 | 18 | 24 | 30 | 36 | 42 |
|  |  |  |  |  |  |  |
| 7 | 14 | 21 | 28 | 35 | 42 | 49 |

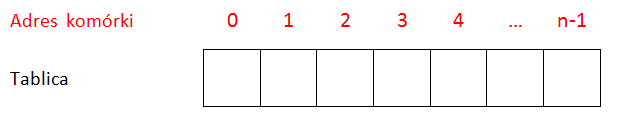
1. Napisz skrypt tworzący tabliczkę mnozenia dla dowolnej $n. np:$n=7

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  |  |  |  |  |  |  |
| 2 | 4 | 6 | 8 | 10 | 12 | 14 |
|  |  |  |  |  |  |  |
| 3 | 6 | 9 | 12 | 15 | 18 | 21 |
|  |  |  |  |  |  |  |
| 4 | 8 | 12 | 16 | 20 | 24 | 28 |
|  |  |  |  |  |  |  |
| 5 | 10 | 15 | 20 | 25 | 30 | 35 |
|  |  |  |  |  |  |  |
| 6 | 12 | 18 | 24 | 30 | 36 | 42 |
|  |  |  |  |  |  |  |
| 7 | 14 | 21 | 28 | 35 | 42 | 49 |

Dział B (9pkt)

Deklaracja tablicy przedstawia się następująco. Nie określamy typu przechowywanych danych, jedynie stwierdzamy, że zmienna tablica będzie… tablicą 🙂

**$tablica** = **array**();

**[](https://tinf.pl/wp-content/uploads/2019/02/js_tablice.png)**

Tablica jest to struktura przechowująca dane jednakowego typu, np. tablica liczb, znaków, itp.

Każda komórka tablicy zawiera jeden element czy strukturę. Każda komórka posiada swój numer.

Numerację komórek tablicy rozpoczynamy od 0. Stąd pierwsza komórka tablicy ma numer 0, druga 1, a dziesiąty, ostatni element w 10-elementowej tablicy ma numer… 9 🙂

Możemy stworzyć tablicę, która od razu będzie coś zawierała. W przypadku liczb wygląda to następująco:

**$tablica** = **array**(1,2,3,4,5,6,7,8,9,10);

Wypiszemy teraz zawartość naszej zmiennej $tablica. Zrobimy to na dwa sposoby. Pierwszy – wykorzystując wbudowaną funkcję print\_r, drugi – tworząc własne wypisanie za pomocą pętli for.

count($tablica) – ilość elementów w tablicy $tablica

print\_r(**$tablica**);

Array ( [0] => 1 [1] => 2 [2] => 3 [3] => 4 [4] => 5 [5] => 6 [6] => 7 [7] => 8 [8] => 9 [9] => 10 )

**for**(**$i**=0; **$i<count**(**$tablica**);**$i++**){

**echo** **$tablica[$i]."** ";

}

1 2 3 4 5 6 7 8 9 10

Stwórzmy teraz tablicę zawierającą stringi – ciągi znaków.

**$zwierzeta** = **array**("kot", "pies", "mysz", "biedronka");

**echo** **$zwierzeta[2]**;

Wypisujemy na ekran zawartość komórki o adresie 2, czyli… ?

Mysz 🙂

Stwórzmy teraz pustą tablicę, wypełnijmy ją wykorzystując mechanizm pętli, następnie wypiszmy jej zawartość na ekran:

**$tab** = **array**();

**for**(**$i**=0; **$i<10**;**$i++**){

**$tab[$i]**=**$i+1**;

}

**for**(**$i**=0; **$i<count**(**$tab**);**$i++**){

**echo** **$tab[$i]."** ";

}

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Numery z dziennika** | | | | | | | | | | | | | | | |
| **NRD** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
|  | **Zadania do realizacji - nr** | | | | | | | | | | | | | | | |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|  | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 1 |
|  | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 1 | 2 | 3 | 4 |
|  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  | 13 | 14 | 15 | 16 | 17 | 18 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  | 16 | 17 | 18 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

1. **Utwórz skrypt**
   1. **Tworzący tablicę o nazwie tab1, zawierającą następujące liczby całkowite: 7, 3, 1, 6, 9, 5, 4, 10, 2, 2.**
   2. **Zmieniający zawartość 7 komórki na 20;**
   3. **Wypisujący zawartość 5 i 7 komórki;**
2. Utwórz skrypt tworzący tablice 20 elementową wypełnioną wartościami losowymi i wypisujący wartości poziomo;
3. Utwórz skrypt tworzący tablice 20 elementową wypełnioną wartościami losowymi i wypisujący wartości pionowo;
4. Utwórz skrypt tworzący tablice 20 elementową wypełnioną wartościami losowymi i znajdujący największy element tablicy
5. Utwórz skrypt tworzący tablice 20 elementową wypełnioną wartościami losowymi i znajdujący najmniejszy element tablicy
6. Utwórz skrypt tworzący tablice 20 elementową wypełnioną wartościami losowymi i obliczający średnią wszystkich elementów
7. Utwórz skrypt tworzący tablice 20 elementową wypełnioną wartościami losowymi i obliczający sumę wszystkich elementów
8. Utwórz skrypt tworzący tablice 20 elementową wypełnioną wartościami losowymi i zwiększający każdy element tablicy o 2;
9. Utwórz skrypt tworzący tablice 20 elementową wypełnioną wartościami losowymi i zwiększający każdy element tablicy o 2;
10. Utwórz skrypt tworzący tablice 20 elementową wypełnioną wartościami losowymi i zwiększający każdy element tablicy razy 3;
11. Utwórz skrypt tworzący tablice 20 elementową wypełnioną wartościami losowymi i obliczający ilość liczb parzystych
12. Utwórz skrypt tworzący tablice 20 elementową wypełnioną wartościami losowymi i obliczający ilość liczb nieparzystych
13. Utwórz skrypt tworzący tablice 20 elementową wypełnioną wartościami losowymi i obliczający ilość liczb podzielnych przez 3.
14. Utwórz skrypt tworzący tablice 20 elementową wypełnioną wartościami losowymi i obliczający ilość sumę parzystych
15. Utwórz skrypt tworzący tablice 20 elementową wypełnioną wartościami losowymi i obliczający ilość sumę nieparzystych
16. Utwórz skrypt tworzący tablice 33 elementową wypełnioną wartościami losowymi i obliczający ilość sumę podzielnych przez 3.
17. Utwórz skrypt tworzący tablice 33 elementową wypełnioną wartościami losowymi i wyświetl na ekran liczby znajdujące się w komórkach o nieparzystych indeksach.
18. Utwórz skrypt tworzący tablice 33 elementową wypełnioną wartościami losowymi i wyświetl na ekran liczby znajdujące się w komórkach o parzystych indeksach.

Dział c (2pkt)

### Pętla do-while

Jest ona bardzo podobna do pętli while. Różnica jest jedna - w do-while wyrażenie sprawdzane jest dopiero po wykonaniu iteracji. W praktyce oznacza to tyle, że iteracja wykona się **zawsze przynajmniej raz**, nawet gdyby wyrażenie nie spełniło się ani razu.

Możliwość użycia niniejszej konstrukcji zależy więc od sytuacji. Z moich obserwacji wynika, że dość rzadko się ją spotyka.

Standardowo zobaczmy najpierw ogólny zapis:

**<?php**

**do {**

**// instrukcje do wykonania w każdej iteracji pętli**

**}**

**while (wyrażenie);**

Ważny jest średnik na końcu, w poprzednich strukturach go nie było.

Zapiszmy przykład z poprzednich lekcji przy użyciu pętli do-while:

**<?php**

**$number = 1;**

**do {**

**echo $number . ' ';**

**$number++;**

**} while ($number <= 10);**

**// result: 1 2 3 4 5 6 7 8 9 10**

Przypomnę, że pierwszy przebieg pętli wykona się, nawet jeśli wyrażenie nigdy nie będzie prawdziwe.

**<?php**

**$number = 50;**

**do {**

**echo $number . ' ';**

**$number += 10;**

**} while ($number > 100);**

**// result: 50**

Czy muszę znów wspominać o alternatywnej składni, nieskończonej pętli i zapisie bez użycia klamer? Szczegóły znajdziesz w dokumentacji.

Dokumentacja: [www.php.net/manual/en/control-structures.do.while.php](https://www.php.net/manual/en/control-structures.do.while.php)

## [Ćwiczenia](https://www.phpdevs.pl/struktury-kontrolne/6-petla-do-while#cwiczenia)

1. Przy użyciu jednej pętli do-while wypisz wszystkie liczby nieparzyste z zakresów 20-40 lub 60-80.