

JS 04 – Szyfrowanie i sortowanie (11)

Szyfrowanie to technika zabezpieczania wrażliwych informacji przed nieautoryzowanym dostępem, a jego podstawowe metody to szyfrowanie symetryczne i asymetryczne, które różnią się sposobem zarządzania kluczami. Szyfrowanie jest wykorzystywane zarówno przez osoby prywatne, jak i organizacje, aby chronić dane takie jak dokumenty finansowe czy informacje osobiste, zapewniając ich poufność. Protokół HTTPS wykorzystuje szyfrowanie do ochrony danych przesyłanych przez internet.

Sortowanie to proces porządkowania danych według określonego kryterium. Istnieje wiele różnych algorytmów sortowania, które można zastosować w zależności od potrzeb i charakterystyki danych. Oto kilka popularnych metod sortowania: **sortowanie bąbelkowe** (Bubble Sort), **sortowanie przez wstawianie** (Insertion Sort), **sortowanie przez wybieranie** (Selection Sort), **sortowanie szybkie** (Quick Sort), **sortowanie przez scalanie** (Merge Sort), **sortowanie kubelkowe** (Bucket Sort), **sortowanie przez zliczanie** (Counting Sort).

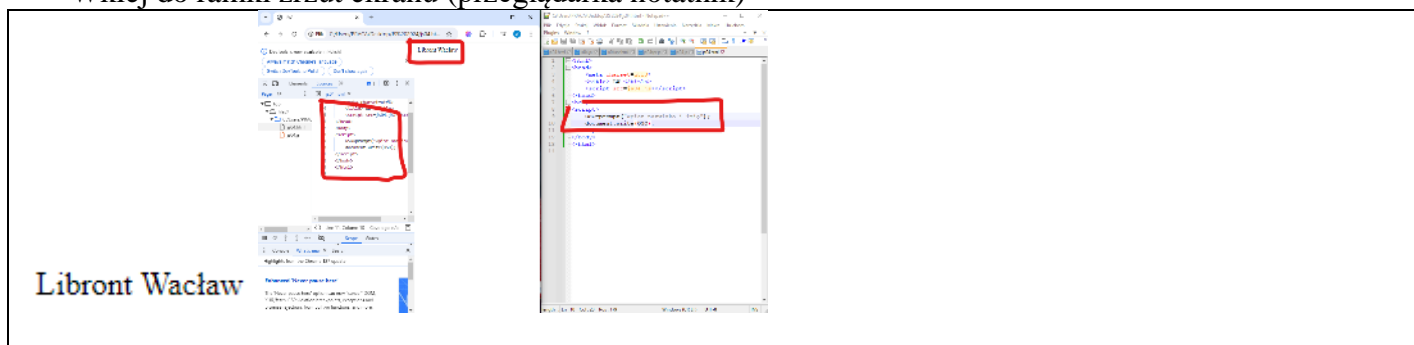
Pamiętaj o tym, by zrzut ekranu DOKUMENTOWAŁ Twoją pracę

Pliki (1)

- W swoim folderze utwórz 2 nowe dokumenty: **js04.html** i **js04.js**
- Otwórz oba dokumenty w notatniku, a dokument HTML w przeglądarce
- Ustaw notatnik i przeglądarkę na połowach ekranu
- Do dokumentu **HTML** wklej tekst z ramki

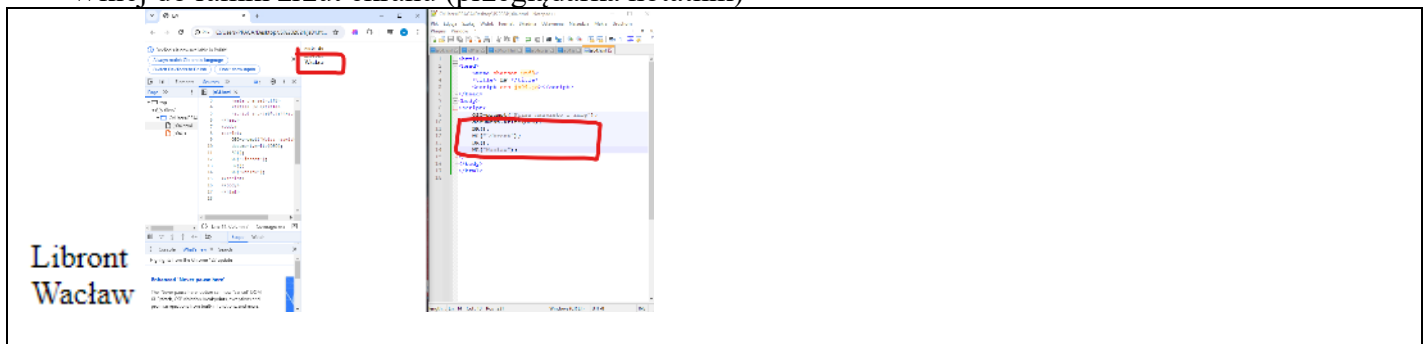
```
<html>
<head>
  <meta charset=utf8>
  <title> SZYFROWANIE </title>
  <script src=js04.js></script>
</head>
<body>
<script>
  OSO=prompt("Wpisz nazwisko i imię");
  documentwrite(oso);
</script>
</body>
</html>
```

- Zmień tytuł strony **SZYFROWANIE** na swoje **inicjały**
- Zapisz dokumenty i odśwież przeglądarkę
- Wpisz swoje nazwisko i imię
plik html zawiera błędy, które należy poprawić za pomocą odpluskwiacza
- Uruchom odpluskwiacz:
 - prawy przycisk myszki w oknie przeglądarki i **Zbadaj**
 - wybierz: **Sources**
 - wybierz plik **js04.js**
- Popraw 2 błędy w jednym wierszu: `documentwrite(oso);`
 - brak kropki
 - nieprawidłowa nazwa zmiennej
- Wklej do ramki zrzut ekranu (przeglądarka notatnik)



Funkcje (1)

- Do dokumentu **JS** wpisz dwie funkcje
`function BR() {document.write("
");}`
`function WR(napis) {document.write(napis);}`
funkcja BR() – wypisuje na stronie znacznik przejścia do nowego wiersza zamiast `document.write(„
”)`;
funkcja WR(napis) – wpisuje na stronie napis zamiast `document.write(napis)`;
- W dokumencie HTML:
 - przejdź do nowego wiersza
funkcja BR()
 - napisz na stronie **swoje nazwisko**
funkcja WR()
 - pod spodem napisz **Twoje imię**
funkcja BR() i WR()
- Zapisz dokumenty i odśwież przeglądarkę
do wyskakującego okienka możesz wpisać cokolwiek
- Wklej do ramki zrzut ekranu (przeglądarka notatnik)



Funkcja Przystawny (1)

Ten typ szyfrowania przestawia znaki w innej kolejności. Najprostszym przykładem takiego szyfrowania jest pisanie wspak. Natomiast funkcja Przystawny() zamienia miejscami dwa kolejne znaki napisu podanego jako parametr.

- Do dokumentu **JS** wklej tekst z ramki

```
function Przystawny(n) {
    var d=n.length;
    var w="";
    for (var i=0;i<d-1;i=i+2){
        var z1=n[i];
        var z2=n[i+1];
        w=w+z2+z1;
    }
    if (d%2==1){
        w=w+n[d-1];
    }
    return w;
}
```

Pętla iteruje w następujący sposób: 0, 2, 4, 6, 8...

Z napisu *N* pobieramy dwa kolejne znaki *Z1[i]* i następny *Z2[i+1]*

Do napisu *W* doklejamy te znaki w zmienionej kolejności

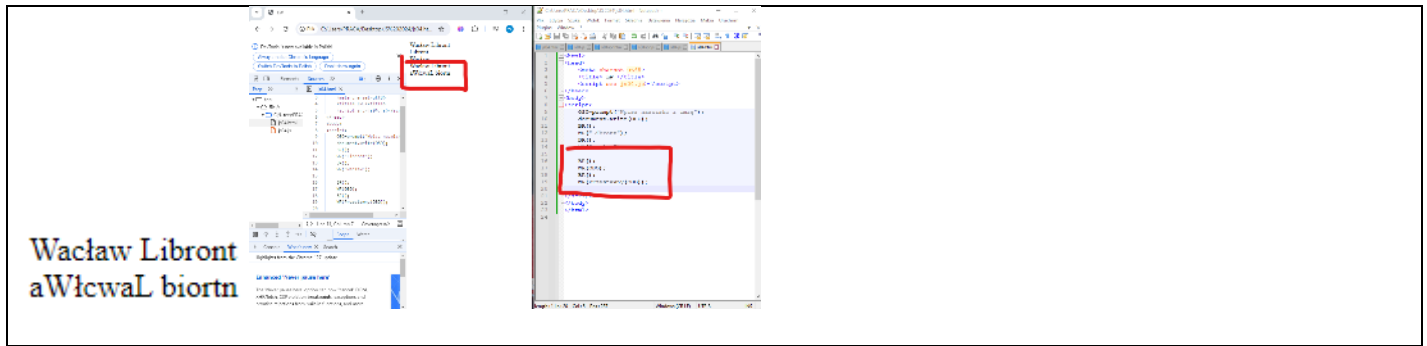
Jeżeli napis *N* ma nieparzystą długość, to

na końcu napisu *W* doklejamy ostatni znak napisu *N*

- Za pomocą funkcji **Przystawny()** zaszyfruj napis ze zmiennej **OSO**

```
BR();
WR(OSO);
BR();
WR(Przystawny(OSO));
```

- Zapisz dokumenty i odśwież przeglądarkę
do wyskakującego okienka wpisz nazwisko i imię
- Wklej do ramki zrzut ekranu (przeglądarka notatnik)



Szyfr Cezara (1)

Szyfr przestawieniowy stosowany już w starożytności, w którym każdy znak w napisie jest zastępowany przez inny, oddalony o stałą liczbę pozycji (tzw. klucz) w zestawie znaków (tzw. alfabecie). Zmienna ALF jest alfabetem i wpisana jest w postaci tekstu i będziesz ją mógł wkleić w swój program.

Nazwa szyfru pochodzi od Juliusza Cezara, rzymskiego wodza i polityka. Szyfrował on prywatną korespondencję do swoich przyjaciół, zapisaną po łacinie, używając szyfru przesuwającego z kluczem 3. W typowych zastosowaniach szyfr korzysta z tzw. tablicy kodów ASCII, my posłużymy się alfabetem w zmiennej ALF, która jest wpisana w postaci tekstu i będziesz ją mógł wkleić w swój program.

- Do dokumentu JS wklej funkcję z ramki

```
function Przesun(znak, klucz) {
    var ALF="AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz";
    var d=ALF.length;
    var poz=ALF.indexOf(znak);
    poz=(poz+klucz)%d;
    if (poz<0) {poz=poz+d;}
    return ALF[poz]
}
function Cezar(n,klucz) {
    w="";
    for (var i=0;i<n.length;i++){
        w=w+Przesun(n[i],klucz);
    }
    return w;
}
```

Funkcja Przesun() przesuwa ZNAK w alfabecie ALF o KLUCZ miejsc

Za pomocą polecenia indexOf() znajduje pozycję znaku w napisie ALF do pozycji POZ dodaje KLUCZ i wykonuje operację modulo

gdy nowa pozycja jest dłuższa niż ALF, to zliczanie od początku alfabetu

jeżeli pozycja jest ujemna, to powiększ o długość alfabetu – uwzględniamy ujemny klucz

funkcja zwraca znak z nowej pozycji alfabetu ALF

Funkcja Cezar przesuwa każdą literę napisu N o liczbę miejsc opisaną w parametrze KLUCZ

pętla FOR wybiera kolejne znaki z napisu

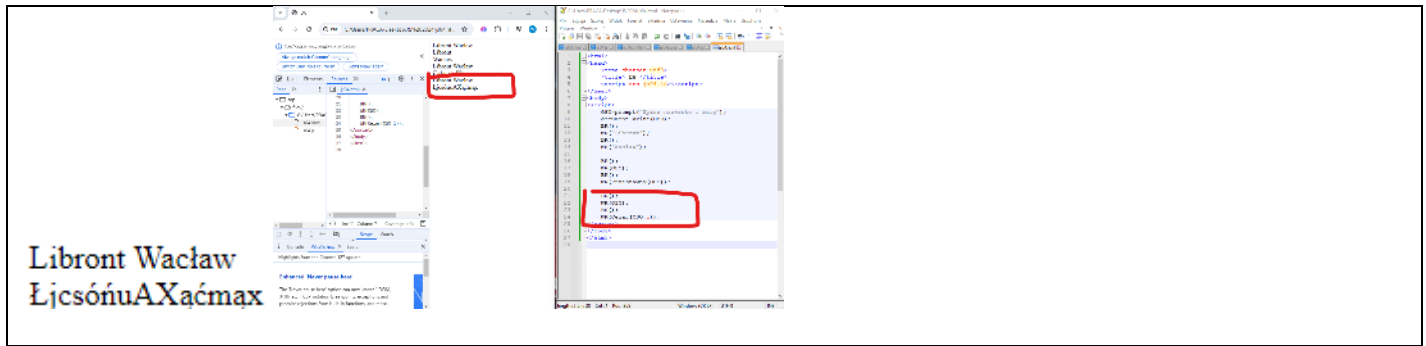
przesunięty znak zostaje „wyliczony” za pomocą funkcji PRZESUN

przesunięte znaki sklejane są w łańcuch tekstowy w zmiennej W

- Za pomocą funkcji Cezar() zaszyfruj napis ze zmiennej OSO z przesunięciem o 1 w alfabecie

```
BR();
WR( );
BR();
WR(Cezar( , ));
```

- Zapisz dokumenty i odśwież przeglądarkę do wyskakującego okienka wpisz nazwisko i imię
- Wklej do ramki zrzut ekranu (przeglądarka notatnik)



Szyfr Parkan (1)

- Do dokumentu JS wklej funkcję z ramki

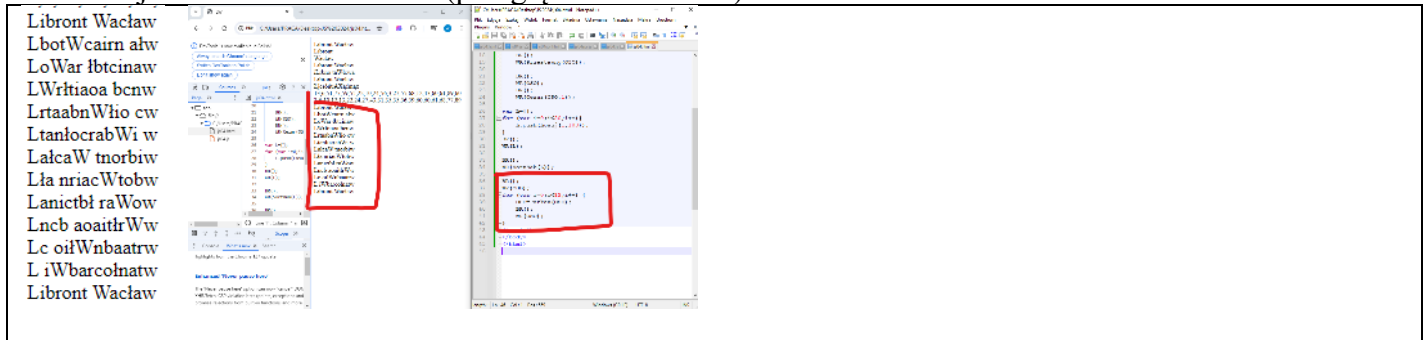
```
function Parkan(napis) {
  var t="";
  var d=napis.length;
  for (var i=0;i<d;i=i+2){
    var z=napis[i];
    t=t+z;
  }
  for (var i=1;i<d;i=i+2){
    var z=napis[i];
    t=t+z;
  }
  return t
}
```

Funkcja Parkan() skleja ze sobą najpierw znaki o parzystych indeksach, a następnie dokleja na koniec znaki o indeksach nieparzystych

- Za pomocą funkcji **Parkan()**
 - zaszyfruj napis ze zmiennej **OSO**
 - pętlę powtórz tyle razy, aby napis powrócił do pierwotnej postaci

```
BR();
WR(OSO);
for (var i=0;i<OSO.length;i++) {
  OSO=Parkan(OSO);
  BR();
  WR(OSO);
}
```

- Wklej do ramki zrzut ekranu (przeładowanie notatnik)



Liczby losowe (1)

Do testowania sortowania potrzebne są nieuporządkowane dane liczbowe. Zamiast wpisywać je ręcznie posłużymy się generatorem liczb losowych. Funkcja Losuj() zwraca losową liczbę z określonego przedziału.

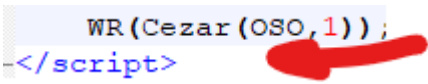
Listy są strukturami danych, które mogą przechowywać wiele wartości tego samego typu. To taka tablica, z komórkami na przykład na liczby.

- Do dokumentu JS wklej funkcję z ramki

```
function Losuj(pocz, koni) {
  return Math.floor(Math.random() * (koni-pocz+1)+pocz);
}
```

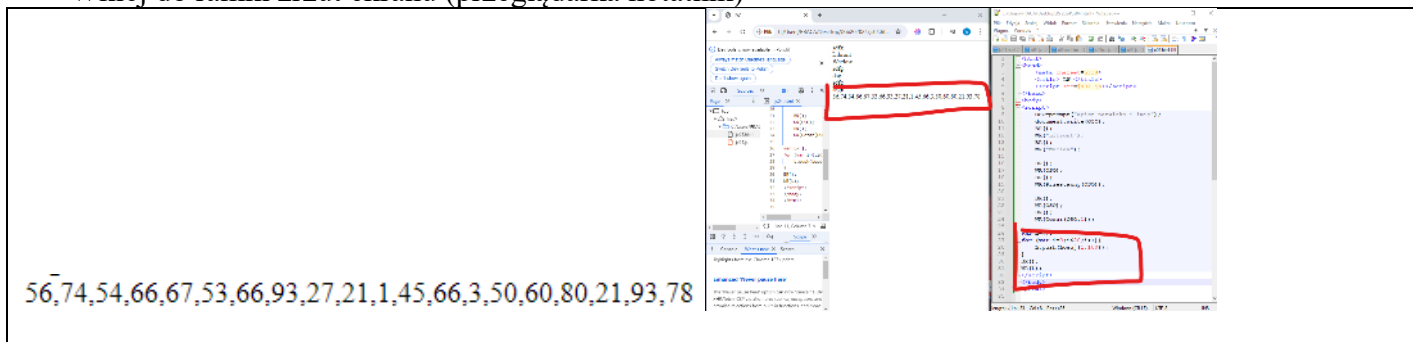
Funkcja **MATH.RANDOM** daje rzeczywistą liczbę losową z przedziału 0..1

liczba jest mnożona przez „rozpiętość” liczb (KONI-POCZ+1) i przesunięta o POCZ
a następnie zamieniana na liczbę całkowitą

- Do dokumentu **HTML**, przed znacznik `</script>`  wpisz tekst

```
var L=[];  
for (var i=0;i<20;i++){  
    L.push(Losuj(1,100));  
}  
BR();  
WR(L);
```

Losujemy 20 liczb z przedziału $<1..100>$ do tablicy L
Deklarujemy tablicę L początkowo pustą []
Za pomocą funkcji PUSH dodajemy nowy element na koniec tablicy
- Zapisz dokumenty i odśwież stronę
- Wklej do ramki zrzut ekranu (przeglądarka notatnik)



Sortowanie bąbelkowe (1)

Sortowanie i wyszukiwanie informacji zajmuje współczesnym komputerom znaczącą część czasu pracy i szybkość algorytmów wpływa znacząco na jakość pracy z programami. Wyobraźmy sobie, gdyby wyszukiwarka Google podawała wynik po 10 minutach w nieuporządkowany sposób!

Sortowanie bąbelkowe jest jednym z najpopularniejszych sposobów sortowania - porównujemy dwa kolejne elementy listy, jeżeli nie są uporządkowane, to zamieniamy je miejscami.

- Do dokumentu **JS** wklej funkcję z ramki

```
function SortBab(lista){  
    var d=lista.length;  
    for (var i=0;i<d;i++){  
        for (var j=0;j<d;j++){  
            if (lista[i]<lista[j]){  
                var l=lista[i];  
                lista[i]=lista[j];  
                lista[j]=l;  
            }  
        }  
    }  
    return lista;  
}
```

Funkcja SortBab sortuje podaną jako parametr listę – tablicę liczb

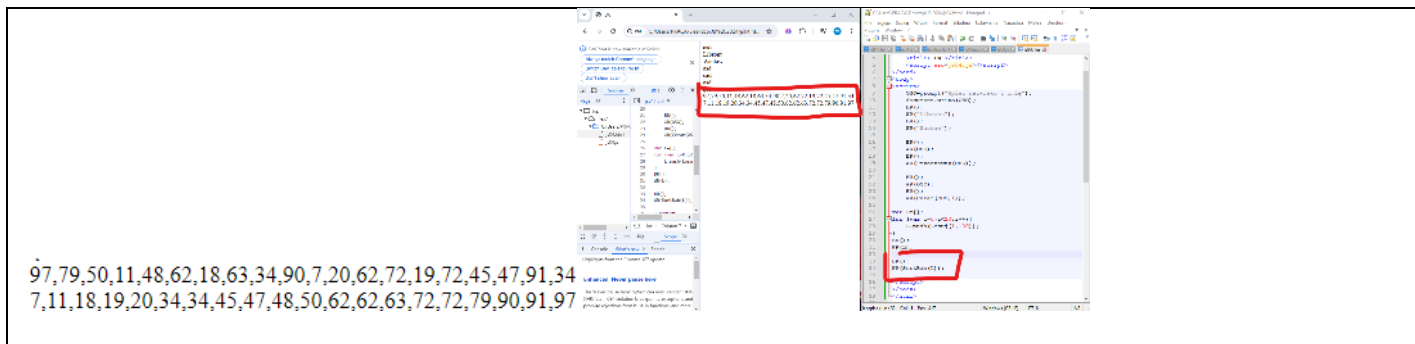
Pętla wewnętrzna (ze zmienną J) porównuje kolejne elementy listy i największy przesuwa na koniec

Pętla zewnętrzna (ze zmienną I) uruchamia pętlę wewnętrzną tyle razy ile elementów ma lista – za każdym razem jedna liczba przesuwana

Zamieniamy miejscami dwa elementy na liście za pomocą dodatkowej zmiennej L

- Za pomocą funkcji **SortBab()** posortuj liczby wylosowane do tablicy L w poprzednim ćwiczeniu

```
BR();  
WR(SortBab(L));
```
- Wklej do ramki zrzut ekranu (przeglądarka notatnik)



Sortowanie tekstów (1)

Funkcja SortBab() potrafi porządkować liczby lub napisy zapisane w osobnych komórkach tablicy, ale nie potrafi posortować wyrazów w zdaniu, albo znaków w napisie.

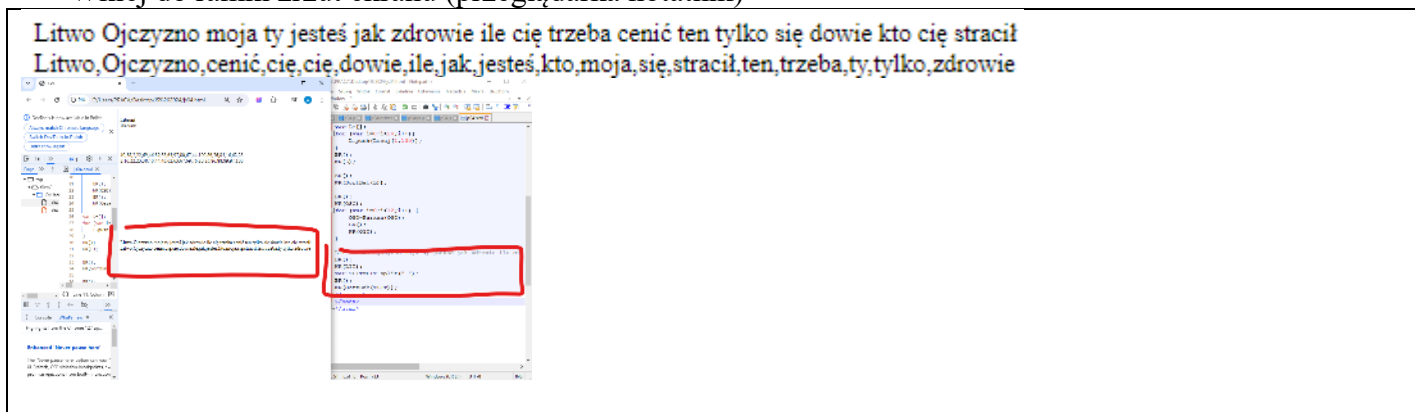
Litwo Ojczyzno moja ty jesteś jak zdrowie ile cię trzeba cenić ten tylko się dowie kto cię stracił

- Za pomocą funkcji SortBab() uporządkuj wyrazy Inwokacji

```
LIT="Litwo Ojczyzno moja ty jesteś jak zdrowie ile cię trzeba cenić ten tylko się dowie kto cię stracił";
BR();
WR(LIT);
var TLIT=LIT.split(" ");
BR();
WR(SortBab(TLIT));
```

do zmiennej LIT przypisz cały tekst z ramki
funkcja split dzieli tekst na wyrazy i zapisuje w tablicy TLIT

- Wklej do ramki zrzut ekranu (przeglądarka notatnik)



Sortowanie znaków (1)

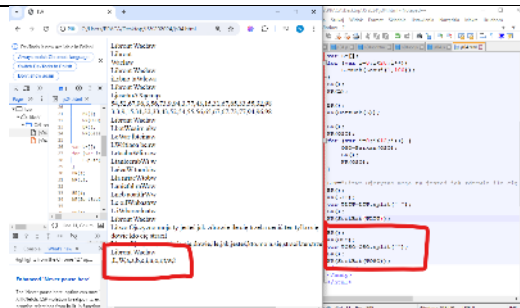
- Za pomocą funkcji SortBab() uporządkuj litery w zmiennej OSO
twoje nazwisko i imię

```
BR();
WR(OSO);
var TOSO=OSO.split("");
BR();
WR(SortBab(TOSO));
```

funkcja split() dzieli zmienną OSO na litery i dlatego tekst pusty w funkcji

- Zapisz dokumenty i odśwież przeglądarkę
do wyskakującego okienka wpisz nazwisko i imię
- Wklej do ramki zrzut ekranu (przeglądarka notatnik)

Libront Waclaw
,L,W,a,a,b,c,i,n,o,r,t,w,l



Szum (1)

Uczniowie na zajęciach z fizyki zapisali „szum”, który wydobywał się z gwiazdy odległej od Ziemi o 1000 lat świetlnych, położonej w gwiazdozbiornie Kasjopei. Okazało się, że „szum” może być zaszyfrowaną informacją pochodzącą od inteligentnych istot. Zastosowano szyfrowanie cezarem i okazało się, że mamy do czynienia z naukową sensacją. Tam są rozumne istoty! Spróbuj odszyfrować „szum” za pomocą funkcji Cezar(). Twoim zadaniem będzie znaleźć klucz. Możesz szukać klucza ręcznie, ale łatwiej będzie wygenerować kolejne wersje w pętli

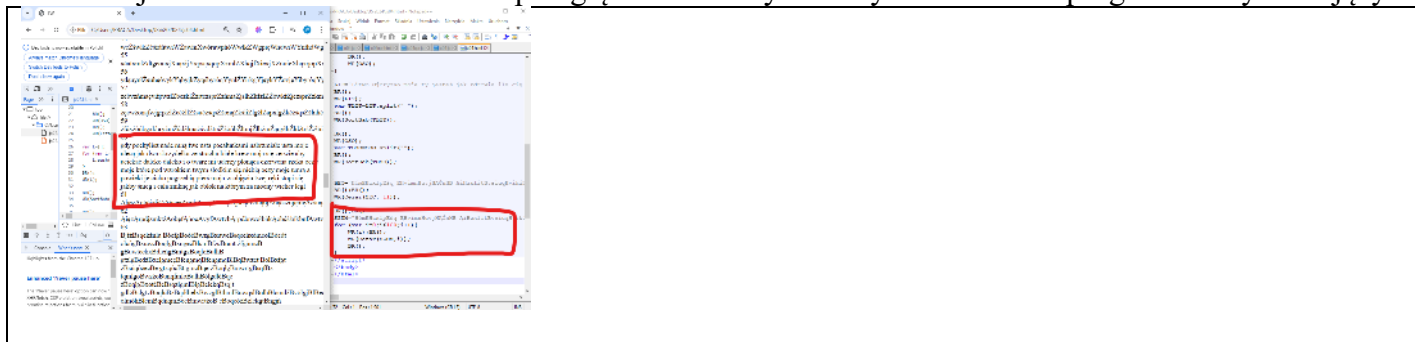
```
HómEHZxlpEśq ĘHvimmHuvjHAĈnHB ĀiHHzxlitBvsiuqHvikžĚuqitnHB ĀiHuxrnHBśnljHrisHmĈiH  
szĚEmntsiHEĚnH AżilpBHKqitnHsznĈHuxriH  
qñHEñžĈqnHikEHBłqnsiĤHmiśnsxHmiśnsxHqHxHAĈižĚHuqHBmnžĚEHztvxvjliHlĚnžĈxviHžĚnsiHxłĚEHuxr  
nHsAyžnHzxmhĈĚžxsqnuHAĈĚuH txmsquH  
qñHvqnkqjHxłĚEHuxrnHBužjHiHžxĈqnsqHrnHlqlpxHžxóžĚnkqjHžqnžAHuxriHĈHxkrñlqBHAĈnrHžñsqH  
AxxzqH qñHriskEHAVqnóHqHlitiHEvqsvñHrisHxktxsHviHsAyžEuHEiHuxlvEHĈqlpnžHśnót
```

- Algorytm programu do odszyfrowania szumu za pomocą szyfru Cezara:
 - przypisz do zmiennej SZUM tekst szumu z ramki
 - pętla FOR „kręci się” od 0 do 100
 - wypisz wartość zmiennej sterującej pętlą
 - wypisz wynik funkcji Cezar SZUM - pierwszy parametr zmienna sterująca - drugi parametr

```
BR();BR();  
SZUM="HómEHZxlpEśq ĘHvimmHuvjH  
for (var i=0;i<100;i++){  
  WR(i);BR();  
  WR(Cezar(SZUM,i));  
  BR();  
}
```

pętla wygeneruje 100 wersji odszyfrowanego tekstu

- Odszukaj odszyfrowany tekst w „ludzkiej mowie”
- Wklej do ramki zrzut ekranu okna przeglądarki z odszyfrowanym wierszem i program odszyfrowujący



Losowa tablica (1)

Na stronie zostanie wypisana tabela 10x10 z dwucyfrowymi losowymi liczbami

- W dokumencie JS napisz funkcję wyświetlającą tablicę liczb według algorytmu:
- Nazwa funkcji LosowaTablica()
- Funkcja wykonuje następujące polecenia:
 - pętla FOR, w której „x” zmienia się od 0 do 9
 - pętla FOR, w której „y” zmienia się od 0 do 9
 - za pomocą funkcji Losuj() wypisz na stronie losowaną liczbę z przedziału 11..99

funkcja losuj() jest już wpisana do pliku

- wypisz na stronie spację

- przejdź do nowego wiersza za pomocą funkcji BR()

```
function LosujTablica () {  
    for (var x=0; x<10; x++) {  
        for (var y=0; y<10; y++) {  
            WR(Losuj(0, 10));  
            WR(" ");  
        }  
        BR();  
    }  
}
```

- W dokumencie HTML uruchom funkcję
BR();
LosowaTablica();
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu JS

The screenshot shows a web browser window with a 10x10 grid of random numbers. The numbers are: 60 43 25 33 93 83 84 70 42 54, 27 26 12 28 86 19 82 42 68 40, 33 62 73 18 73 12 98 33 20 80, 15 54 97 98 57 97 83 72 31 17, 77 80 65 25 69 35 67 83 79 30, 38 12 49 95 38 45 54 59 78 52, 64 56 37 24 47 45 78 35 30 17, 86 47 42 16 66 61 21 45 97 12, 76 29 81 52 60 18 62 58 32 36, 29 26 78 29 93 14 62 36 23 95. The browser's developer console shows the JavaScript code: function LosujTablica () { for (var x=0; x<10; x++) { for (var y=0; y<10; y++) { WR(Losuj(0, 10)); WR(" "); } BR(); } } and the output of the function, which is the grid of numbers. The console also shows the function call: LosowaTablica();