

Zadanie: tekst → HTML


- Program przyjmujący przez wiersz polecenia
 - nazwę pliku wejściowego
 - nazwę pliku wyjściowego

np. `mojprogram plik.txt nowyplik.html`

- Do nowego pliku program powinien wypisać
 - `<html><body>`
 - treść pliku tekstowego, przy czym pomiędzy kolejnymi liniami ma się znaleźć tag `
`
 - `</body></html>`

Liczby pseudolosowe


- Funkcja rand
 - zwraca losową liczbę całkowitą z zakresu od 0 do RAND_MAX



```
#include <stdlib.h>
```

Liczby pseudolosowe

- Funkcja rand
 - zwraca losową liczbę całkowitą z zakresu od 0 do RAND_MAX
- Liczba z dowolnego zakresu?
 - `rand() % N`
zwraca liczbę z zakresu od 0 do N-1



```
#include <stdlib.h>
```

Liczby pseudolosowe

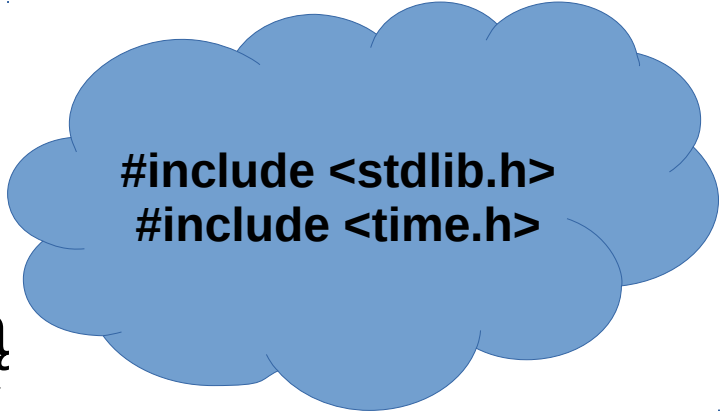
- Funkcja rand
 - zwraca losową liczbę całkowitą z zakresu od 0 do RAND_MAX
- Liczba z dowolnego zakresu?
 - `rand() % N`
zwraca liczbę z zakresu od 0 do N-1
- Wylosujmy sobie liczbę z zakresu od 1 do 36.



```
#include <stdlib.h>
```

Liczby pseudolosowe

- Funkcja rand
 - zwraca losową liczbę całkowitą z zakresu od 0 do RAND_MAX
- Liczba z dowolnego zakresu?
 - `rand() % N`
zwraca liczbę z zakresu od 0 do N-1
- Wylosujmy sobie liczbę z zakresu od 1 do 36.
 - Wszyscy dostają to samo?
 - Dopiszmy linię wyżej
`srand(time(0));`



```
#include <stdlib.h>
#include <time.h>
```

Ćwiczenie 1.

- Napisz program, który wylosuje 6 liczb z zakresu od 1 do 36.
 - Można wydzielić część kodu w funkcję generującą tablicę N liczb z zakresu od 1 do M

Ćwiczenie 2.

- Wybierz 6 liczb z zakresu od 1 do 36 (ulubione liczby, data urodzenia, etc.)
- Napisz program, który wielokrotnie
 - wylosuje 6 liczb z zakresu od 1 do 36,
 - sprawdzi, czy wylosowane zostały wybrane liczby
- Wyznacz, za którym powtórzeniem wylosujemy nasze „wybrane” liczby.

Ćwiczenie 2.

- Wybierz 6 liczb z zakresu od 1 do 36 (ulubione liczby, data urodzenia, etc.)
- Napisz program, który wielokrotnie
 - wylosuje 6 liczb z zakresu od 1 do 36,
 - **sprawdzi, czy wylosowane zostały wybrane liczby**
- Wyznacz, za którym powtórzeniem wylosujemy nasze „wybrane” liczby.
- **PROBLEM: jak porównać ze sobą dwie nieuporządkowane tablice?**

Sortowanie

- Sortowanie = układanie elementów w kolejności
- Jeden z prostszych algorytmów sortowania:
sortowanie bąbelkowe (*bubble sort*)

Bubble Sort Example

9, 6, 2, 12, 11, 9, 3, 7
6, 9, 2, 12, 11, 9, 3, 7
6, 2, 9, 12, 11, 9, 3, 7
6, 2, 9, 12, 11, 9, 3, 7
6, 2, 9, 11, 12, 9, 3, 7
6, 2, 9, 11, 9, 12, 3, 7
6, 2, 9, 11, 9, 3, 12, 7
6, 2, 9, 11, 9, 3, 7, 12

Ćwiczenie 2.

- Wybierz 6 liczb z zakresu od 1 do 36 (ulubione liczby, data urodzenia, etc.)
- Napisz program, który wielokrotnie
 - wylosuje 6 liczb z zakresu od 1 do 36,
 - sprawdzi, czy wylosowane zostały wybrane liczby
- Wyznacz, za którym powtórzeniem wylosujemy nasze „wybrane” liczby.
 - Można wydzielić część kodu w funkcję sortującą tablicę N liczb

Benchmark

- Sprawdźmy, ile trwa sortowanie bąbelkowe
- Stwórzmy program, który
 - Utworzy tablicę 100000 losowych liczb
 - Pobierze aktualny czas $t_0 = \text{time}(0)$;
 - Uruchomi funkcję sortującą
 - Pobierze aktualny czas $t_1 = \text{time}(0)$;
 - Sprawdź, czy tablica jest posortowana
 - Wypisze czas sortowania $t_1 - t_0$.

Quicksort

- Biblioteka języka C zawiera implementację quicksort

function

qsort

```
void qsort (void* base, size_t num, size_t size,  
            int (*compar)(const void*,const void*));
```

Parameters

base

Pointer to the first object of the array to be sorted, converted to a void*.

num

Number of elements in the array pointed to by *base*.

`size_t` is an unsigned integral type.

size

Size in bytes of each element in the array.

`size_t` is an unsigned integral type.

compar

Pointer to a function that compares two elements.

This function is called repeatedly by qsort to compare two elements. It shall

```
int compar (const void* p1, const void* p2);
```

Benchmark

- Sprawdźmy, ile trwa sortowanie „quicksort”
- Stwórzmy program, który
 - Utworzy tablicę 100000 losowych liczb
 - Pobierze aktualny czas $t_0 = \text{time}(0)$;
 - Uruchomi funkcję sortującą
 - Pobierze aktualny czas $t_1 = \text{time}(0)$;
 - Sprawdzi, czy tablica jest posortowana
 - Wypisze czas sortowania $t_1 - t_0$.