Student: Marija Erić

Zadatak: 1

Datum: 7.11.2025.

## 1. Opis zadatka

Traži se da izvršavamo funkciju dobijanja broja Pi Leibnitzevom metodom korišćenjem WASM(u C) i JS i uporedimo vreme izvršavanja u zavisnosti od broja iteracija.

## 2. Plan rada

Znači prvo upalim Xampp i pokrenem Apache server, zatim udjem u emsdk folder i pokrenem njegov cmd emcmdprompt.bat, zatim idem u svoj folder za zadatak u vs codu, napravim pi.c fajl gde stavim funkciju za izračunavanje PI Leibnitzevom metodom, zatim pravim template.html koji sadrži JS funkciju za Pi, poziv C funkcije putem Module.ccall, input polje za broj iteracija, dugme za uporedjivanje. Kad to završim, iz emscriptena pokrećem emcc komandu koja pravi pi.html, pi.js i pi.wasm i na kraju sve testiram preko browsera lokalno.

## 3. Beleške tokom rada

Prvo Xampp i start Apache. Zatim u vs codu cmd kucam

C:\Users\ja\Downloads\RAF\Sistemsko\emsdk> .\emcmdprompt.bat

Zatim C:\xampp\htdocs\wasm\zadatak1>

Zatim pi.c double pi\_leibniz(int n) {

if (n <= 0) return 0.0;

double s = 0.0, sign = 1.0;

for (int k = 0; k < n; ++k) {

s += sign / (2.0 \* k + 1.0);

sign = -sign;

}

return 4.0 \* s;

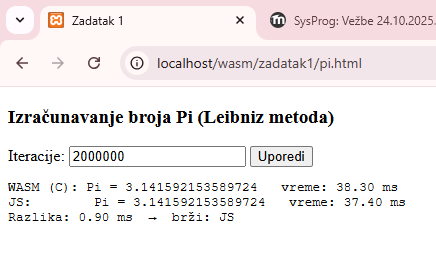
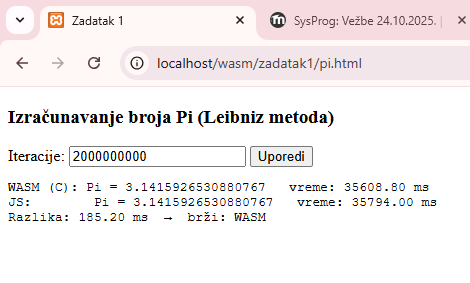
}

Zatim template.html: u <script> delu stavljam JS funkciju piLeibnizJS(n) sa istom petljom kao u C, pomoćne funkcije za ispis, element sa idjem output, objekat Module sa onRuntimeInitialized, dugme btnUporedi i u click handleru uzimam n iz <input id="n">, prvo merim vreme i pozivam C funkciju:  
const t0w = performance.now();  
const valW = Module.ccall('pi\_leibniz','number',['number'],[n]);  
const t1w = performance.now();  
a zatim merim vreme i pozivam JS verziju. Na kraj ispisujem oba rezultata i vreme i računam razliku.

Pravim pi.html pi.js i pi.wasm koristeci emcc pi.c -O3 -o pi.html --shell-file template.html -s NO\_EXIT\_RUNTIME=1 -s EXPORTED\_FUNCTIONS=[\_pi\_leibniz] -s EXPORTED\_RUNTIME\_METHODS=ccall

I na kraju pokrecem u browseru http://localhost/wasm/zadatak1/pi.html

## 4. Rezultat



## 5. Zaključak

Kroz ovaj zadatak sam naučila kako da funkciju u C kompajliram u WASM i pozovem u JS, medjutim zaključak je da nije uvek WASM brži. Konkretno sam zaključila da ako imamo recimo 2000000 iteracija i manje JS radi brže (zbog manjeg overheada), a tek ako imamo 200000000 iteracija WASM jasno pobedjuje. Program mislim da radi kako treba.

## 6. Rešenje

Pokreće se tako što prvo treba pokrenuti XAMPP i startovati Apache server, zatim otvoriti u browseru adresu: [http://localhost/zadatak1/pi.html](http://localhost/zadatak01/pi.html) i uneti broj iteracija i kliknuti na dugme “Uporedi”.