Fortran 2008
Projekt 1

1 Ogólne zasady zaliczenia

1. Zadanie powinno być napisane w Fortranie 2008
2. Kod powinien być kompilowany za pomocą cmake - konfiguracja do pobrania tutaj - określa ona jednocześnie obowiązującą strukturę plików w projekcie
3. Dopuszczalne jest używanie kompilatora ifort z odpowiednią opcją kompilacji wymuszającą standard 2008
4. Projekt powinien być opisany w pliku README.md (proszę o używanie składni Markdown)
5. Postęp prac nad projektem powinien być wersjonowany przy zachowaniu adekwatnych opisów commitów
6. Podstawę do zaliczenia projektu stanowi wysłany w terminie link do pobrania repozytorium z projektem
7. Zakazane jest używanie CVS, Team Foundation Server oraz płatnych systemów wersjonowania
8. Repozytoria mogą być hostowane w dowolnym miejscu (np. Github, Bitbucket, prywatny serwer itp.) pod warunkiem udzielenia odpowiedniego dostępu do kodu
9. Za serwer daty/godziny przyjmuje się serwer poczty AGH
10. Projekty powinny wykorzystywać jak największą ilość poznanych na zajęciach mechanizmów składni języka przy zachowaniu dobrych praktyk programistycznych

2 Opis problemu

Poniższe kody źródłowe napisane są w języku Julia.
Oryginalna treść dostępna tutaj - K. Rycerz, MOwNiT, lab 3.

2.1 Naiwne mnożenie macierzy

Mnożenie macierzy - wersja naiwna.

```
function naive_multiplication(A,B)
C=zeros(Float64,size(A,1),size(B,2))
  for i=1:size(A,1)
    for j=1:size(B,2)
        for k=1:size(A,2)
            C[i,j]=C[i,j]+A[i,k]*B[k,j]
        end
    end
end
C
end
```

2.2 Poprawione mnożenie macierzy

Poprawiona funkcja korzytająca z powyższego oraz z faktu, że można zmieniać kolejność operacji dodawania (a co za tym idzie kolejność pętli).

```
function better_multiplication( A,B )
C=zeros(Float64,size(A,1),size(B,2))
  for j=1:size(B,2)
    for k=1:size(A,2)
        for i=1:size(A,1)
            C[i,j]=C[i,j]+A[i,k]*B[k,j]
        end
    end
end
C
end
```

2.3 Iloczyn skalarny

Kolejna poprawiona funkcja wykorzystuje z powyższego oraz faktu, że mnożenie wiersza przez kolumnę (lub na odwrót) można przedstawić jako iloczyn skalarny. Przerób powyższe (poprawione, sekcja 2.2) mnożenie macierzy tak, aby wykorzytać funkcję **dot_product** wbudowaną w Fortran.

3 Zadanie

- Proszę zaimplementować trzy powyższe algorytmy mnożenia macierzy w Fortranie. **UWAGA** - w Fortran-ie macierz przechowywana jest kolumnami (odwrotnie niż w C!).
- Należy porównać działanie czterech algorytmów - trzech powyższych oraz **matmul** wbudowanego w Fortran-a.
- Przedstawić dla każdej precyzji wyniki na jednym wykresie - pliki **wykres4.pdf, wykres8.pdf, wykres16.pdf**, wykonanym w **gnuplot** z następującymi opcjami

```
unset grid
set terminal pdf
set output 'wykres[kind].pdf'
set key box top left
set multi
set logscale x
set logscale y
set key opaque
set key box
set key width 1 height 0.5  font "Arial, 14"
set style data lines
set termopt enhanced
set xlabel "N" font "Arial, 14"
set ylabel "multiplication time" font "Arial, 14"
set xtics  font "Arial, 14"
set ytics  font "Arial, 14"
set termoption dashed

plot [pierwszy wykres]....
replot [kolejny wykres]...

unset multi
```

- Pomiary mają być zrobione dla macierzy kwadratowych 10×10, 20×20, 40×40, $\ldots$, 1280×1280.

3.1 Struktura kodu i plików

- W module naivemath (plik src/naivemath.F90) powinna znajdować się naiwna matoda mnożenia macierzy - **function naivmull**.

```
function naivmull (A, B) res(C)
implicit none
real (kind=4), intent(in), dimension(:,:) :: A,B
```

- W module bettermath (plik src/bettermath.F90) powinna znajdować się poprawiona matoda mnożenia macierzy - **function bettmull**.

```
function bettmull (A, B) res(C)
implicit none
real (kind=4), intent(in), dimension(:,:) :: A,B
```

- W module dotmath (plik src/dotmath.F90) powinna znajdować się poprawiona matoda mnożenia macierzy - **function dotmull**.

```
function dotmull (A, B) res(C)
implicit none
real (kind=4), intent(in), dimension(:,:) :: A,B
```

- Proszę samemu dopasować optymalny typ zwracany przez w.w. funkcje.
- Funkcje **naivmull**, **bettmull** oraz **dotmull** powinny być interfejsami ukrywającymi (prywatne dla modułu) osobne wersje mnożenia dla typów **real (kind=4,8,16)**.
- W pliku src/main.F90 ma być główny program.
- W katalogu res/ mają być wyniki pomiarów, pliki gnuplot-a oraz pdf-y.
- W pliku README.md powinny znajdować się osadzone (embedded) w.w. wykresy razem z wnioskami.

4 Termin i sposób nadsyłania rozwiązań

Za ostateczny termin oddania zadania uznaje się piątek 26.04.2019, godzinę 23:59:59. Za każdy rozpoczęty dzień spóźnienia odejmowany jest jeden punkt.
Zadania wysyłane są na adres mailowy **macwozni@agh.edu.pl**. Temat wiadomości powinien być następującej postaci: **[FORTRAN] Nazwisko Imie Zad1** - np. **[FORTRAN] Kowalski Jan Zad1**.
Niepoprawnie zaadresowane rozwiązania nie zostaną ocenione (0 pkt.).