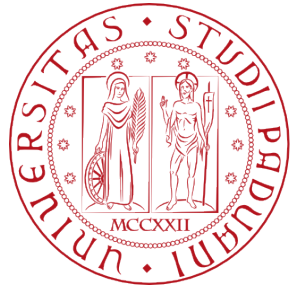
Quantum computation and computing 2022-2023 Assignment Week 1 - Basic

Nguyen Xuan Tung - 2005491

Nov 2022



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Setup

- The first program is to print out the name and surname when we type them in the terminal.
- FirstName, LastName and OtherName are character variables that can hold a string of no more than 20 characters: character*20 :: FirstName, LastName
- Trim function is used to remove trailing blank characters of a string

```
PS D:\Physics-Data-Msc\Quantum-IC\Assignment-1> gfortran -o setup setup.f90
PS D:\Physics-Data-Msc\Quantum-IC\Assignment-1> ./setup
 What's your name?
Tung
Nguyen
 Hello Tung Nguyen
```

Number precision

- Integer*2 values range from -32,768 to 32,767 and are stored in 2 contiguous bytes so the result returned error.
- An INTEGER*4 occupies 4 bytes.which can store 65536*65536 distinct values. This gives a range of approximately 4 billion different values. So the result returned correct.

```
D:\Physics-Data-Msc\Quantum-IC\Assignment-1> gfortran -o 2a 2a.f90 -fno-range-check
D:\Physics-Data-Msc\Quantum-IC\Assignment-1> ./2a
um of 2.000.000 and 1 with INTEGER*2 using 2 bytes: -31615
he result of Sum of 2.000.000 and 1 with INTEGER*2 using 2 bytes  is wrong!!!
um of 2.000.000 and 1 with INTEGER*2 using 4 bytes:     2000001
he result of Sum of 2.000.000 and 1 with INTEGER*2 using 4 bytes  is correct!!!
```

Number precision

- Real*4 specifies the variable names to be single precision 4-byte real numbers which has 7 digits of accuracy and a magnitude range of 10 from -38 to +38 (represent for single precision).
- Real*8 specifies the variable names to be double precision 8-byte real numbers which has 15 digits of accuracy and a magnitude range of 10 from -308 to +308 (represent for double precision).
- we can see the result with double precision is more accurate compare with the result with single precision.

```
PS D:\Physics-Data-Msc\Quantum-IC\Assignment-1> gfortran 2b.f90 -o 2b -fno-range-check
PS D:\Physics-Data-Msc\Quantum-IC\Assignment-1> ./2b
 Sum pi * 10^32 and sqtr(2) * 10^21 with single precisi
```

Performance testing

- We write function to fill the matrix with random variables
- the matrix dimension stored in a vector of two dimensions: N, M
- In the function performs matrix multiplication:
  - the element of the matrix are given in input: ii, jj, kk
  - a matrix C is given as a result of product of matrix A and B
- We also check that if the multiplication is possible
- The results is printed in the terminal which requests the performer to type the dimension of the matrix in order to calculate the matrix multiplication

Performance testing

- delta time is created with real*8 in order to calculate the cpu time for every matrix multiplication
- we will perform 3 different matrix multiplication and check the needed to perform them
- the result shows different time need to compute each matrix multiplication

```
PS D:\Physics-Data-Msc\Quantum-IC\Assignment-1> gfortran -o 3c 3c.f90
PS D:\Physics-Data-Msc\Quantum-IC\Assignment-1> ./3c
 Time passed using matrix multiplication   5.0937500000000000      s
 Time passed using transposed matrix multiplication   4.0937500000000000      s
 Time passed using transposed matrix multiplication   6.2500000000000000E-002 s
```