1. (5) Write a \LaTeX\ script which generates the following equations with the given format:

\[
\begin{align*}
  x &= r \cos \theta, \\
  y &= r \sin \theta.
\end{align*}
\]

Solution

\begin{verbatim}
\begin{eqnarray*}
  x &=& r \cos \theta, \\
  y &=& r \sin \theta.
\end{eqnarray*}
\end{verbatim}

2. (10) Give the output of the following Fortran statements:

(a) \texttt{print*, 1/2*2}
(b) \texttt{print*, 1/2.*2.}
(c) \texttt{print*, 1./2.*2}
(d) \texttt{print*, 1./2./2.}
(e) \texttt{print*, 1.8/1}

Solution

(a) 0
(b) 1.000000
(c) 1.000000
(d) 0.2500000
(e) 1.000000000000000
3. (5) The hexadecimal system is a base 16 system. It first sixteen numbers are

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f.

Their base 10 equivalents are

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.

Represent the base 10 number 64 in hexadecimal.

Solution

40 = 4 \times 16^1 + 0 \times 16^0.

4. (10) Convert the following mathematical expression into Fortran code

\[ x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}. \]

Solution

\[ x = (-b+sqrt(b**2-4.*a*c))/2./a \]

5. (5) Write the UNIX commands for

(a) creating a directory named mydirectory,
(b) removing a directory named mydirectory.

Solution

(a) mkdir mydirectory
(b) rmdir mydirectory

TURN THE PAGE!
6. (20) Carl Friedrich Gauss (1777-1855) while in grade school was asked to add all of the integers between 1 and 100 (including 1 and 100). Write a Fortran code to achieve this end. You only need to write the code; however, if you are also able to report the numerical value (as Gauss famously and quickly did without the benefit of a computer) of the result, there will be a special award.

**Solution**

```fortran
program test
    implicit none
    integer :: sum, i
    sum = 0
    do i=1,100
        sum = sum+i
    enddo
    print*,sum
end program test
```

Gauss recognized the sum was also equal to

\[
\frac{n(n+1)}{2} = \frac{100 \times 101}{2} = 5050.
\]

Running the code verifies Gauss’ answer.

7. (10) Give the output of the following program

```fortran
program test
    integer :: i, j, k
    do i=2,8,2
        do j=i,2
            do k = 1,j,2
                print*,i,j,k
            enddo
        enddo
    enddo
end program test
```

**Solution**

```
2 2 1
```

8. (30) Consider a matrix \( \mathbf{A} \) of dimension \( N \times M \) and a vector \( \mathbf{x} \) of dimension \( M \). All elements of \( \mathbf{A} \) and \( \mathbf{x} \) are to be real double precision numbers. Write a complete Fortran code that
(a) reads from the screen the variable array dimension \( N \) and \( M \),
(b) reads from the screen each of the elements of \( \mathbf{x} \),
(c) reads from the screen each of the elements of \( \mathbf{A} \),
(d) builds a vector \( \mathbf{b} \) formed by the matrix multiplication \( \mathbf{b} = \mathbf{A} \cdot \mathbf{x} \).
(e) prints each element of \( \mathbf{b} \) to the screen.

Use a \texttt{do} loop to compute \( \mathbf{A} \cdot \mathbf{b} \). Do not use \texttt{matmul}.

---

\textit{Solution}

```fortran
program test
    implicit none
    integer :: n, m, i, j
    real (kind=8), dimension(:,,:), allocatable :: a
    real (kind=8), dimension(:), allocatable :: b, x
    print*, 'enter n'
    read*, n
    print*, 'enter m'
    read*, m
    allocate(a(n,m))
    allocate(x(m))
    allocate(b(n))
    do i=1, n
        do j=1, m
            print*, 'enter a(', i, ',j, ')' 
            read*, a(i,j)
        enddo
    enddo
    do j=1, m
        print*, 'enter x(', j, ')' 
        read*, x(j)
    enddo
    do i=1, n
        b(i) = 0. _8
        do j=1, m
            b(i) = b(i) + a(i,j)*x(j)
        enddo
    enddo
    print*, b
end program test
```

9. (5) Write a short Fortran program which prints to the screen

\textbf{Go Irish! Beat Trojans!}

Have the program also print to the screen an estimate of the final score.

---

\textit{Solution}

```fortran
4
```
program test
print*, 'Go Irish! Beat Trojans!'
print*, 'ND 38, USC 14'
end program test