```
PROGRAM binding_energies

!
! Purpose:
!
!      This program explores the accuracy of the semi-empirical mass formula
!      using model calculations of several hundred nuclei and their various isotopes.
!      Calculations give rise to implications of the shell model structure in nuclei.
!

! ------ Variables Declarations ------ !

IMPLICIT NONE
INTEGER :: i, k, j
INTEGER, DIMENSION(5000) :: Z, N, A, m_atomic1, m_atomic2
REAL, DIMENSION(5000) :: m_atomic, Binding, Semi, Diff, Diff_per_A, Diff_per_Z, Diff_pe
REAL, DIMENSION(5000) :: Binding_per_A, SEMF_per_A, E_coul, E_surf, E_sym, E_vol, Error
REAL, DIMENSION(5000) :: dB_dA_odd, dB_dA_NZeven, dB_dA_NZodd, d_pairing, y, x
REAL :: pairing_A, BE, SEMF

! ------ Constant Declarations ------ !

REAL, PARAMETER :: m_N = 1.008665 ! Neutron mass in MeV/c^2 (u)
REAL, PARAMETER :: m_Z = 1.007825 ! Proton mass in MeV/c^2 (u)
REAL, PARAMETER :: c_squared = 931.5 ! conversion factor for atomic mass unit
REAL, PARAMETER :: a_vol = 15.5 ! MeV
REAL, PARAMETER :: a_surf = -16.8 ! MeV
REAL, PARAMETER :: a_coul = -0.72 ! MeV
REAL, PARAMETER :: a_sym = -23 ! MeV

! ------ Input/Output Files ----- !

OPEN ( UNIT=10, FILE='nuclear.dat', STATUS='UNKNOWN' )
OPEN ( UNIT=20, FILE='mass.dat', STATUS='OLD' )
OPEN ( UNIT=30, FILE='energies.dat', STATUS='UNKNOWN' )

! ------ Graphs of binding energies vs. Z/N/A ------ !

OPEN ( UNIT=40, FILE='bindingenergies_vs_A.dat', STATUS='UNKNOWN' )
OPEN ( UNIT=50, FILE='bindingenergies_vs_Z.dat', STATUS='UNKNOWN' )
OPEN ( UNIT=60, FILE='bindingenergies_vs_N.dat', STATUS='UNKNOWN' )
OPEN ( UNIT=80, FILE='junk.dat', STATUS='UNKNOWN' )

OPEN ( UNIT=101, FILE='Error.dat', STATUS='UNKNOWN' )
OPEN ( UNIT=201, FILE='Line_of_stability.dat', STATUS='UNKNOWN' )
OPEN ( UNIT=301, FILE='y_vs_x.dat', STATUS='UNKNOWN' )

! ------ Graphs of derivatives ------ !

OPEN ( UNIT=102, FILE='dB_dZ.dat', STATUS='UNKNOWN' )
OPEN ( UNIT=103, FILE='dB_dN.dat', STATUS='UNKNOWN' )
OPEN ( UNIT=104, FILE='dB_dA_odd.dat', STATUS='UNKNOWN' )
OPEN ( UNIT=105, FILE='dB_dA_NZeven.dat', STATUS='UNKNOWN' )
OPEN ( UNIT=106, FILE='dB_dA_NZodd.dat', STATUS='UNKNOWN' )


!      OPEN ( UNIT=31, FILE='stable_energies.dat', STATUS='UNKNOWN' )
!      OPEN ( UNIT=41, FILE='bindingenergies_vs_stableA.dat', STATUS='UNKNOWN' )
!      OPEN ( UNIT=51, FILE='bindingenergies_vs_stableZ.dat', STATUS='UNKNOWN' )
!      OPEN ( UNIT=61, FILE='bindingenergies_vs_stableN.dat', STATUS='UNKNOWN' )

!      OPEN ( UNIT=32, FILE='single_energies.dat', STATUS='UNKNOWN' )
!      OPEN ( UNIT=42, FILE='bindingenergies_vs_singleA.dat', STATUS='UNKNOWN' )
```

```
!      OPEN ( UNIT=52, FILE='bindingenergies_vs_singleZ.dat', STATUS='UNKNOWN' )
!      OPEN ( UNIT=62, FILE='bindingenergies_vs_singleN.dat', STATUS='UNKNOWN' )

! ------ Creating graphs for individual energies as a function of A ------ !

OPEN ( UNIT=33, FILE='Diff_per_A.dat', STATUS='UNKNOWN' )
OPEN ( UNIT=43, FILE='Diff_per_Z.dat', STATUS='UNKNOWN' )
OPEN ( UNIT=53, FILE='Diff_per_N.dat', STATUS='UNKNOWN' )

! ------ Graphs of Binding Energy per A vs. A for classical and SEMF ------ !

OPEN ( UNIT=44, FILE='BE_per_A_vs_A.dat', STATUS='UNKNOWN' )
OPEN ( UNIT=54, FILE='SEMF_per_A_vs_A.dat', STATUS='UNKNOWN' )

! ------ Graphs of individual energy terms per nucleon ------ !

OPEN ( UNIT=35, FILE='E_vol_vs_A.dat', STATUS='UNKNOWN' )
OPEN ( UNIT=45, FILE='E_surf_vs_A.dat', STATUS='UNKNOWN' )
OPEN ( UNIT=55, FILE='E_coul__A.dat', STATUS='UNKNOWN' )
OPEN ( UNIT=65, FILE='E_sym_vs_A.dat', STATUS='UNKNOWN' )

! ------ Code ------ !

! READ loop to obtain values of N, Z, A, and atomic mass
! and format correctly for Fortran
!
! Data acquired from tabulated masses from NNDC

i = 1
100   read (20,300, end=200) N(i), Z(i), A(i), m_atomic1(i), m_atomic2(i)
300   format (6x,i3,2x,i3,2x,i3,77x,i3,1x,i6)

write (10,400) N(i), Z(i), A(i), m_atomic1(i), m_atomic2(i)
400   format (i3,2x,i3,2x,i3,5x,i3,'.',i6.6)

i = i + 1
GO TO 100

200   CONTINUE

CLOSE(10); CLOSE(20);



OPEN ( UNIT=10, FILE='nuclear.dat', STATUS='OLD' )


! Loop to calculate the binding energies using classical formula
! and SEMF, as well as computing the difference in each formula

j = 1
800   read (10, 900, end=1000) N(j), Z(j), A(j), m_atomic(j)
900   format (i3,2x,i3,2x,i3,5x,f10.6)


! Even-oddness algorithm to compute the pairing term, delta(A)
! using modular division


IF ( MOD(Z(j),2) == 0 .AND. MOD(N(j),2) == 0 ) THEN

pairing_A = 34.
```

```
ELSE IF ( MOD(Z(j),2) == 1 .AND. MOD(N(j),2) == 1 ) THEN

pairing_A = -34.

ELSE IF ( MOD(A(j),2) == 1 ) THEN

pairing_A = 0.

END IF


! Classical picture of nucleon numbers !



write(201,*) Z(j),N(j)


! Calculating theoretical binding energy, binding energy
! via the SEMF, and the difference between the two


Binding(j) = BE(m_atomic(j), N(j), Z(j))
Binding_per_A(j) = Binding(j)/float(A(j))

Semi(j) = SEMF(A(j),Z(j)) + pairing_A*A(j)**(-3./4.)
SEMF_per_A(j) = Semi(j)/float(A(j))


! Graphing Binding Energy per A vs. A
! and SEMF per A vs. A

write(44,*) A(j), Binding_per_A(j)
write(54,*) A(j), SEMF_per_A(j)


! Calculating binding energy per nucleon


!       BE_per_A = Binding(j)/float(A(j))
!       SEMF_per_A = Semi(j)/float(A(j))


! Discarding negative (non-physical) BEs
! For graphing all nuclei in the NDCC table

IF ( Binding(j) < 0. .OR. Semi(j) < 0. ) THEN

write (80,*) Binding(j), Semi(j)

ELSE

Diff(j) = Binding(j) - Semi(j)

write(30,*) Binding(j), Semi(j)
write(40,*) A(j), Diff(j)
write(50,*) Z(j), Diff(j)
write(60,*) N(j), Diff(j)

Diff_per_A(j) = Diff(j)/float(A(j))
write(33,*) A(j), Diff_per_A(j)
```

```
Diff_per_Z(j) = Diff(j)/Float(Z(j))
write(43,*) Z(j), Diff_per_Z(j)

Diff_per_N(j) = Diff(j)/float(N(j))
writE(53,*) N(j), Diff_per_Z(j)

Error(j) = Diff(j)/Binding(j)
write(101,*) A(j), Error(j)

! Graphing derivative of BEs w.r.t. Z and N per nucleon !

dB_dZ(j) = a_coul*(2*Z(j)-1)*A(j)**(-1./3.) - 2*a_sym*(N(j)-Z(j))/float(A(j))
write(102,*) Z(j), dB_dZ(j)

dB_dN(j) = 2*a_sym*(N(j)-Z(j))/float(A(j))
write(103,*) N(j), dB_dN(j)

END IF


! Graphing derivatives w.r.t. A !

dB_dA_odd(j) = a_vol + (2./3.)*a_surf*A(j)**(-1./3.) - (1./3.)*a_coul*Z(j)*(Z(j)  &
- 1)*A(j)**(-4./3.) - a_sym*(N(j)-Z(j))**2/A(j)**2

IF ( MOD(A(j),2) == 1 ) THEN

dB_dA_odd(j) = dB_dA_odd(j)
write(104,*) A(j), dB_dA_odd(j)

ELSE IF ( MOD(Z(j),2) == 1 .OR. MOD(N(j),2) == 1 ) THEN

dB_dA_NZodd(j) = dB_dA_odd(j) + 0.75*34.*A(j)**(-7./4.)
write(106,*) A(j), dB_dA_NZodd(j)

ELSE IF ( MOD(Z(j),2) == 0 .OR. MOD(N(j),2) == 0 ) THEN

dB_dA_NZeven(j) = dB_dA_odd(j) - 0.75*34.*A(j)**(-7./4.)
write(105,*) A(j), dB_dA_NZeven(j)

END IF



! Graphing nuclei with N greater than or equal Z (stable)


IF ( Binding(j) < 0. .OR. Semi(j) < 0. ) THEN

write (80,*) Binding(j), Semi(j)

ELSE IF ( N(j) .GE. Z(j) ) THEN

Diff(j) = Binding(j) - Semi(j)

write(31,*) Binding(j), Semi(j)
write(41,*) A(j), Diff(j)
write(51,*) Z(j), Diff(j)
write(61,*) N(j), Diff(j)

END IF
```

```
! Graphing single (stable) nuclei


IF ( N(j) .GE. Z(j) .AND. A(j) .NE. A(j-1) ) THEN

Diff(j) = Binding(j) - Semi(j)

write(32,*) Binding(j), Semi(j)
write(42,*) A(j), Diff(j)
write(52,*) Z(j), Diff(j)
write(62,*) N(j), Diff(j)

END IF


! Graphing energy differences per nucleon !


IF ( Z(j) == 0 .OR.  N(j) == 0 ) THEN

write(80,*) Z(j), N(j)

ELSE

Diff_per_Z(j) = Diff(j)/float(Z(j))
write(43,*) Z(j), Diff_per_Z(j)

Diff_per_N(j) = Diff(j)/float(N(j))
write(53,*) N(j), Diff_per_N(j)

END IF


! Graphing error in SEMF !


! Graphing Individual energy terms per nucleon !


E_vol(j) = a_vol
E_surf(j) = a_surf*A(j)**(-1./3.)
E_coul(j) = Z(j)*(Z(j)-1)*A(j)**(-4./3.)
E_sym(j) = (N(j)-Z(j))**2 / float(A(j))**2

write(35,*) A(j), E_vol(j)
write(45,*) A(j), E_surf(j)
write(55,*) A(j), E_coul(j)
write(65,*) A(j), E_sym(j)

j = j + 1

GO TO 800


1000  CONTINUE

DO WHILE ( k < 200 )

x(k) = k
y(k) = x(k)
write(301,*) x(k), y(k)

k = k + 1
```

```
END DO


CLOSE(30); CLOSE(40); CLOSE(50); CLOSE(60);
CLOSE(31); CLOSE(41); CLOSE(51); CLOSE(61);
CLOSE(32); CLOSE(42); CLOSE(52); CLOSE(62);
CLOSE(33); CLOSE(43); CLOSE(44); CLOSE(54);
CLOSE(35); CLOSE(45); CLOSE(55); CLOSE(65);


STOP

END PROGRAM binding_energies


! ------ Functions ------!


! Function to compute binding energy of each isotope


FUNCTION BE(nuclear_mass,N,Z)

IMPLICIT NONE
REAL :: BE, nuclear_mass
INTEGER :: N, Z

REAL, PARAMETER :: c_squared = 931.5 ! conversion mass factor
REAL, PARAMETER :: m_N = 1.008665 ! MeV/c^2
REAL, PARAMETER :: m_H = 1.007825 ! MeV/c^2

BE = ( float(N)*m_N + float(Z)*m_H - nuclear_mass )*c_squared

RETURN;

END FUNCTION BE


! Function to compute Binding energy using semi-empirical mass formula


FUNCTION SEMF(A,Z)

IMPLICIT NONE
REAL :: SEMF
INTEGER :: A, Z

REAL, PARAMETER :: a_vol = 15.5 ! MeV
REAL, PARAMETER :: a_surf = -16.8 ! MeV
REAL, PARAMETER :: a_coul = -0.72 ! MeV
REAL, PARAMETER :: a_sym = -23 ! MeV

SEMF = a_vol*A + a_surf*A**(2./3.) + a_coul*Z*(Z-1)*A**(-1./3.) + a_sym*( (A-2*Z)**2 )/

RETURN;

END FUNCTION SEMF
```