



Εισαγωγή στον Προγραμματισμό Η/Υ (Fortran 90/95/2003)

Ενότητα 9: Διαδικασίες

Νίκος Καραμπετάκης
Τμήμα Μαθηματικών



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

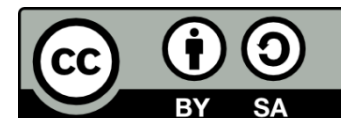


ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Περιεχόμενα Ενότητας

1. Τι είναι διαδικασία.
2. Πως συντάσσεται μία διαδικασία.
3. Αναδρομική διαδικασία.
4. Τι είναι τα modules.
5. Δημιουργία στατικής βιβλιοθήκης.
6. Συναρτήσεις – Διαδικασίες της αριθμητικής IMSL βιβλιοθήκης της Compaq Visual Fortran.
7. Διαδικασία μετάφρασης/εκτέλεσης προγραμμάτων με συναρτήσεις από την IMSL βιβλιοθήκη.



Σκοποί Ενότητας

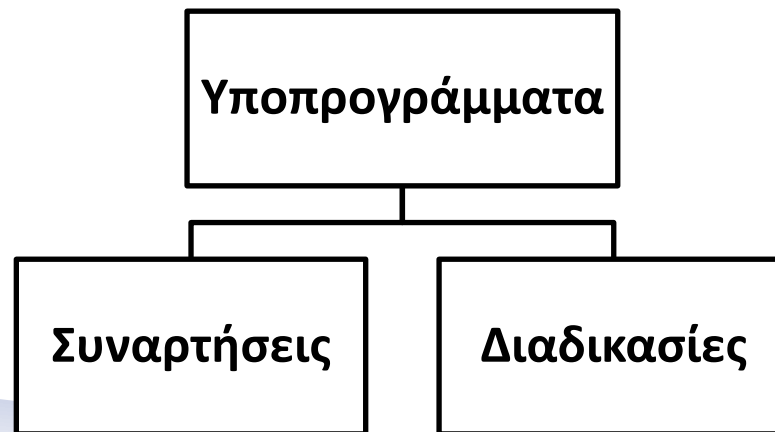
1. Η παρουσίαση του ορισμού και των τρόπων δήλωσης των διαδικασιών.
2. Η παρουσίαση των μεθόδων εισαγωγής δεδομένων/ επιστροφής τιμών σε διαδικασίες.
3. Η παρουσίαση των modules.
4. Η παρουσίαση του τρόπου δημιουργίας στατικής βιβλιοθήκης.
5. Η παρουσίαση της διαδικασίας μετάφρασης/ εκτέλεσης προγραμμάτων με συναρτήσεις από την IMLS βιβλιοθήκη.



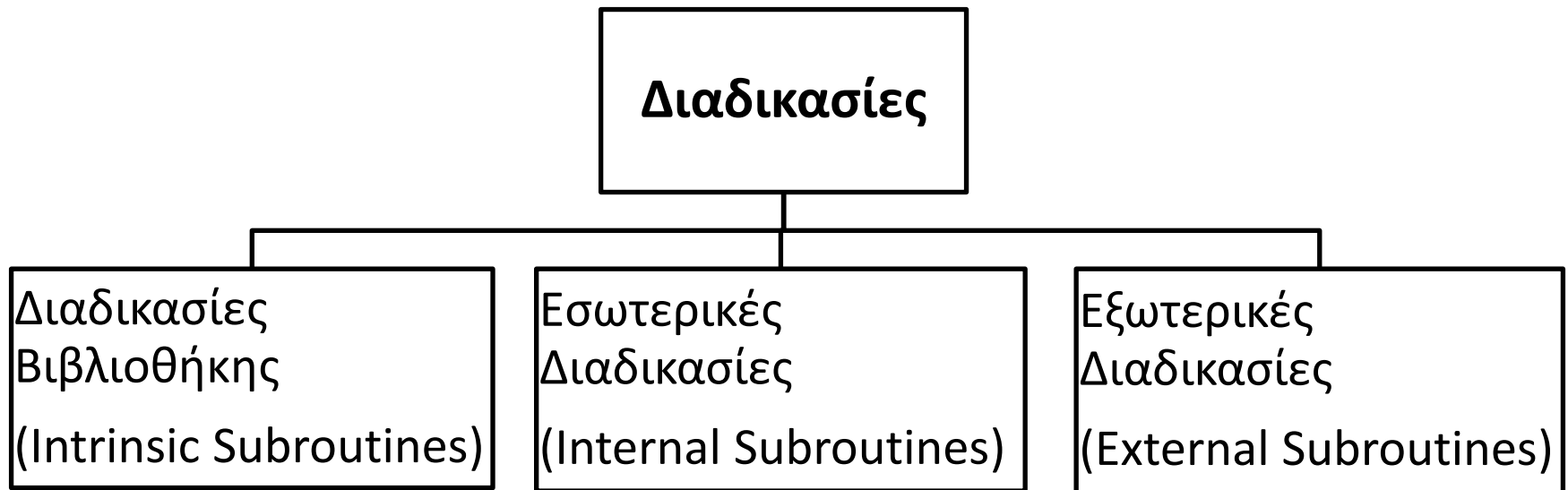
Διαχωρισμός των υποπρογραμμάτων σε δύο κατηγορίες

Συνάρτηση (*function*) : είναι ένα υποπρόγραμμα που στόχο του έχει να υπολογίζει και να επιστρέφει μόνο μια τιμή (αριθμητική, χαρακτήρα, λογική) με το όνομά της όπως οι γνωστές μαθηματικές συναρτήσεις, π.χ. $SIN(X)$, $COS(X)$ κ.λ.π.

Διαδικασία (*subroutine*) : είναι ένα υποπρόγραμμα το οποίο έχει όλα τα χαρακτηριστικά ενός προγράμματος.



Διαχωρισμός των διαδικασιών σε κατηγορίες



Διαδικασία – Σύνταξη

SUBROUTINE «όνομα υπορουτίνας»[(λίστα μεταβλητών)].

IMPLICIT NONE.

Τμήμα Δηλώσεων.

Τμήμα Εκτέλεσης.

Τμήμα Υποπρογραμμάτων.

END SUBROUTINE «όνομα υπορουτίνας» .

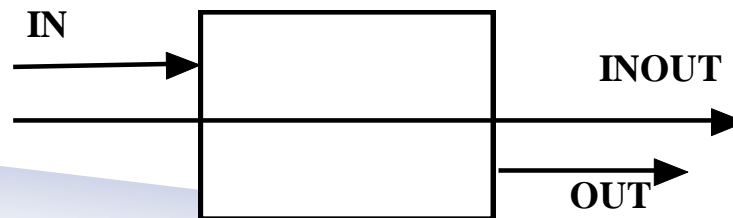


Διαχωρισμός των τυπικών παραμέτρων

INTENT(IN) : δήλωση τυπικών παραμέτρων οι οποίες εισάγουν δεδομένα από το κυρίως πρόγραμμα, χωρίς να αλλάζει η τιμή τους μέσα στη διαδικασία.

INTENT(OUT) : δήλωση τυπικών παραμέτρων οι οποίες εξάγουν αποτελέσματα που υπολογίστηκαν μέσα στη διαδικασία.

INTENT(INOUT) : δήλωση τυπικών παραμέτρων οι οποίες εισάγουν δεδομένα από το κυρίως πρόγραμμα, αλλά και επιστρέφουν αποτελέσματα μέσω της διαδικασίας (και συνεπώς αλλάζει η τιμή τους μέσα στη διαδικασία).



Πως καλείται η διαδικασία από το κυρίως πρόγραμμα;

CALL «όνομα υπορουτίνας»[(λίστα σταθερών, μεταβλητών ή εκφράσεων)].



Παράδειγμα 1

Να γραφεί διαδικασία με το όνομα SWAP, που να μεταθέτει τις τιμές δύο **πραγματικών** αριθμών A, B.

PROGRAM SUBROUTINES1

IMPLICIT NONE

REAL:: A,B

READ*,A,B

CALL SWAPREAL(A,B)

PRINT*,A,B

CONTAINS



Παράδειγμα 1 (...συνέχεια)

```
SUBROUTINE SWAPREAL(X,Y)
```

```
  IMPLICIT NONE
```

```
  REAL, INTENT(INOUT)  :: X,Y
```

```
  REAL                  :: S
```

```
  S=X
```

```
  X=Y
```

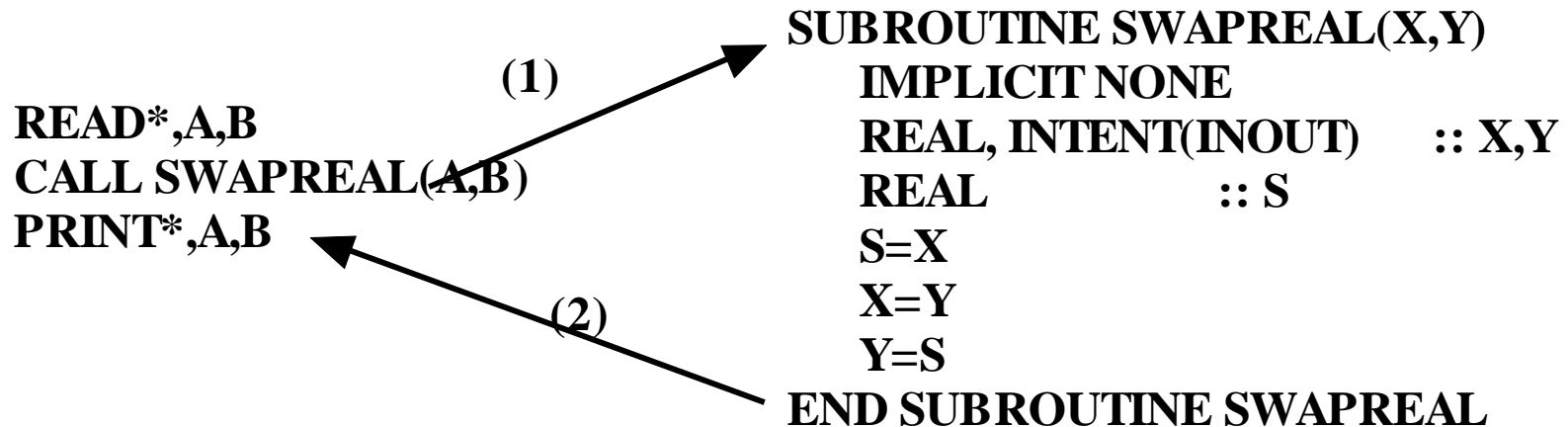
```
  Y=S
```

```
END SUBROUTINE SWAPREAL
```

```
END PROGRAM SUBROUTINES1
```



Σειρά εκτέλεσης εντολών



Μια άλλη λύση μέσω εξωτερικής διαδικασίας – Αρχείο A με εξωτερικές διαδικασίες

```
SUBROUTINE SWAPREAL(X,Y)  
  IMPLICIT NONE  
  REAL, INTENT(INOUT)  :: X,Y  
  REAL                  :: S  
  S=X  
  X=Y  
  Y=S  
END SUBROUTINE SWAPREAL
```



Μια άλλη λύση μέσω εξωτερικής διαδικασίας – Αρχείο A με εξωτερικές διαδικασίες συνέχεια

```
SUBROUTINE SWAPINT(X,Y)
```

```
  IMPLICIT NONE
```

```
  INTEGER, INTENT(INOUT)      :: X,Y
```

```
  INTEGER                      :: S
```

```
  S=X
```

```
  X=Y
```

```
  Y=S
```

```
END SUBROUTINE SWAPINT
```



Μια άλλη λύση μέσω εξωτερικής διαδικασίας – Αρχείο B με το κυρίως πρόγραμμα

PROGRAM SUBROUTINES1

IMPLICIT NONE

INTERFACE SWAP

 SUBROUTINE SWAPREAL(X,Y)

 REAL, INTENT(INOUT) :: X,Y

 END SUBROUTINE SWAPREAL

 SUBROUTINE SWAPINT(X,Y)

 INTEGER, INTENT(INOUT) :: X,Y

 END SUBROUTINE SWAPINT

END INTERFACE SWAP

REAL :: A,B

INTEGER :: M,N

READ*,A,B ; CALL SWAP(A,B) ; PRINT*,A,B

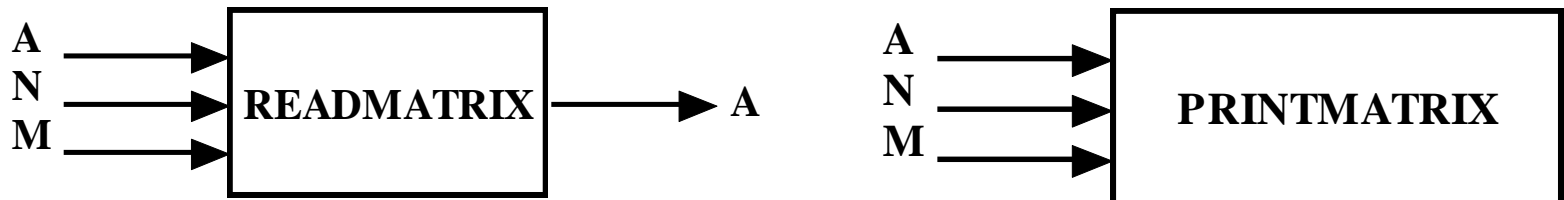
READ*,M,N ; CALL SWAP(M,N) ; PRINT*,M,N

END PROGRAM SUBROUTINES1



Παράδειγμα 2

Να γραφούν δύο διαδικασίες, μια που θα διαβάσει και μια που θα εκτυπώνει έναν πραγματικό πίνακα έκτασης $/N,M/$.



Πρόγραμμα

PROGRAM SUBROUTINES2

IMPLICIT NONE

REAL, DIMENSION(:,:), ALLOCATABLE :: X

INTEGER :: K,L

PRINT*, 'DIMENSIONS ='

READ*, K,L

ALLOCATE (X(K,L))

CALL READMATRIX(X,K,L)

CALL PRINTMATRIX(X,K,L)

CONTAINS

SUBROUTINE READMATRIX(A,N,M)

IMPLICIT NONE

INTEGER, INTENT(IN) :: N,M

REAL, DIMENSION(N,M) , INTENT(INOUT) :: A

INTEGER :: I,J



Πρόγραμμα (...συνέχεια)

```
        DO I=1,N
          READ*,(A(I,J),J=1,M)
        END DO
END SUBROUTINE READMATRIX
SUBROUTINE PRINTMATRIX(A,N,M)
  IMPLICIT NONE
  INTEGER, INTENT(IN) :: N,M
  REAL, DIMENSION(N,M) , INTENT(IN) :: A
  INTEGER                :: I
  DO I=1,N
    PRINT*,A(I,:)
  END DO
END SUBROUTINE PRINTMATRIX
END PROGRAM SUBROUTINES2
```



Άσκηση 1

Να γραφεί εξωτερική διαδικασία που θα δέχεται ως είσοδο δύο πίνακες X , Y τάξης 1 και έκτασης N , και θα υπολογίζει τους συντελεστές A , B της ευθείας γραμμικής παλινδρόμησης ($y=Ax+B$) από τον τύπο :

$$A = \frac{\sum_{i=1}^N X_i \sum_{i=1}^N Y_i - N \sum_{i=1}^N X_i Y_i}{\left(\sum_{i=1}^N X_i\right)^2 - N \sum_{i=1}^N (X_i)^2}$$

$$B = \frac{\sum_{i=1}^N X_i \sum_{i=1}^N X_i Y_i - \sum_{i=1}^N (X_i)^2 \sum_{i=1}^N Y_i}{\left(\sum_{i=1}^N X_i\right)^2 - N \sum_{i=1}^N (X_i)^2}$$



Αναδρομική διαδικασία υπολογισμού N!

Να γραφεί αναδρομική διαδικασία υπολογισμού του N!.

PROGRAM RECSUBS

IMPLICIT NONE

INTEGER :: N,F

READ*,N ; CALL FACTORIAL(N,F) ; PRINT*,F

CONTAINS

RECURSIVE SUBROUTINE FACTORIAL(N,P)

IMPLICIT NONE

INTEGER, INTENT(IN) :: N

INTEGER, INTENT(INOUT) :: P



Αναδρομική διαδικασία υπολογισμού N! (...συνέχεια)

```
IF (N==0) THEN
```

```
    P=1
```

```
ELSE
```

```
! Υπολογισμός του (N-1)! και καταχώρηση του στο P
```

```
    CALL FACTORIAL(N-1,P)
```

```
    P=P*N
```

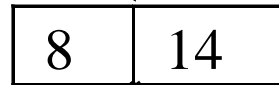
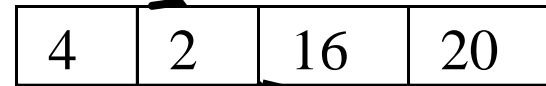
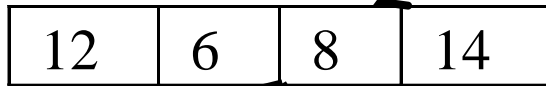
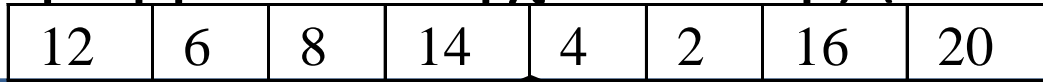
```
END IF
```

```
END SUBROUTINE FACTORIAL
```

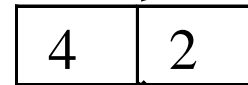
```
END PROGRAM RECSUBS
```



Ταξινόμηση μέσω συγχώνευσης (Merge Sort)



και



&



&



&



&



&



Συγχώνευση Πινάκων

```
SUBROUTINE MERGE(A,N,B,M,C)  
IMPLICIT NONE  
INTEGER, INTENT(IN) :: N,M  
REAL, DIMENSION(N), INTENT(IN) :: A  
REAL, DIMENSION(M), INTENT(IN) :: B  
REAL, DIMENSION(N+M), INTENT(INOUT) :: C  
INTEGER :: I,J,K  
    I=1 ; J=1 ; K=1  
DO WHILE ((I<=N).AND.(J<=M))  
    IF (A(I)<B(J)) THEN  
        C(K)=A(I)  
        I=I+1
```



Συγχώνευση Πινάκων (...συνέχεια)

ELSE

$C(K)=B(J)$

$J=J+1$

END IF

$K=K+1$

END DO

IF (I>N) THEN

$C(K:N+M)=B(J:M)$

ELSE

$C(K:N+M)=A(I:N)$

END IF

END SUBROUTINE MERGEA



Αναδρομική διαδικασία *MergeSort*(A, N)

Βήμα 1. Ορίζουμε τον πραγματικό πίνακα A που θα ταξινομηθεί ως (INOUT) καθώς και την ακέραια μεταβλητή N ως (IN).

Βήμα 2. Αν $N > 1$ τότε
$$M \leftarrow \left\lfloor \frac{N+1}{2} \right\rfloor$$

Βήμα 2α. Υπολογίζουμε την θέση M του μεσαίου στοιχείου του πίνακα.

Βήμα 2β. Καλούμε την διαδικασία *MergeSort*(A(1:M), M) για να ταξινομήσουμε το πρώτο μισό του πίνακα A.

Βήμα 2γ. Καλούμε την διαδικασία *MergeSort*(A(M+1:N), N-M) για να ταξινομήσουμε το υπόλοιπο μισό του πίνακα A.



Αναδρομική διαδικασία *MergeSort*(A, N)

Βήμα 2δ. Καλούμε την διαδικασία *Mergea*($A(1:M), M, A(M+1:N), N-M, C$) προκειμένου να συγχωνεύσουμε τους δύο ήδη ταξινομημένους πίνακες $A(1,M)$ και $A(M+1,N)$ στον πίνακα C .

Βήμα 2ε. Τοποθετούμε τον βοηθητικό πίνακα C στην θέση του A δηλ. $A \leftarrow C$

Διαφορετικά

Βήμα 2'α Επιστρέφουμε στο κυρίως πρόγραμμα.

τέλος αν.



Αναδρομική διαδικασία

RECURSIVE SUBROUTINE MERGESORT(A,N)

IMPLICIT NONE

INTEGER, INTENT(IN) :: N

INTEGER :: M

REAL, DIMENSION(N), INTENT(INOUT) :: A

REAL, DIMENSION(N) :: C

IF (N>1) THEN

M=(N+1)/2

CALL MERGESORT(A(1:M),M)

CALL MERGESORT(A(M+1:N),N-M)

CALL MERGE(A(1:M),M,A(M+1:N),N-M,C)

A=C

END IF

END SUBROUTINE MERGESORT



Αναδρομική διαδικασία (...συνέχεια)

```
PROGRAM MERGE_SORT
  IMPLICIT NONE
  ! Variables
  REAL, DIMENSION(8) :: A
  DATA A/12,6,8,14,4,2,16,20/
  ! Body of program
  CALL MERGESORT(A,8)
  PRINT*, A
  CONTAINS
    SUBROUTINE MERGE(A,N,B,M,C)
    .....
  END SUBROUTINE MERGE
  RECURSIVE SUBROUTINE MERGESORT(A,N)
  .....
  END SUBROUTINE MERGESORT
END PROGRAM MERGE_SORT
```



Σύγκριση Πολυπλοκότητας Αλγορίθμων Ταξινόμησης

$$t(n) \in O(n \log_2 n)$$

n	$t(n) \in O(n \log_2 n)$	$t(n) = n^2$
10	33	100
100	664	10.000
100	9.965	1.000.000
10.000	132.877	100.000.000
100.000	1.660.964	10.000.000.000
1.000.000	19.931.568	1.000.000.000.000
10.000.000	232.534.966	100. 000.000.000.000



Τι είναι τα modules;

Τα modules έχουν δομή σχεδόν ίδια με του προγράμματος. Αποτελούν μια συλλογή από δηλώσεις σταθερών ή/και μεταβλητών, interface μπλοκ, συναρτήσεις και διαδικασίες, τα οποία είναι απαραίτητα σε ένα πρόγραμμα ή σε ένα υποπρόγραμμα, module κλπ. Δεν περιέχουν όμως εκτελέσιμες εντολές (παρά μόνο εντός υποπρογραμμάτων) και γι' αυτό το λόγο δεν μπορούν να υπάρχουν από μόνα τους, αλλά πρέπει να χρησιμοποιούνται από άλλα modules και από το κυρίως πρόγραμμα.

MODULE «όνομα module»

IMPLICIT NONE

[Τμήμα δηλώσεων]

← Σύνταξη

CONTAINS

[Τμήμα συναρτήσεων-διαδικασιών]

END MODULE



Παράδειγμα 3

MODULE EXAMPLE

IMPLICIT NONE

! ΔΗΛΩΣΕΙΣ ΣΤΑΘΕΡΩΝ, ΜΕΤΑΒΛΗΤΩΝ

REAL, PARAMETER :: PI=3.14

REAL, PARAMETER :: G=9.8

TYPE STUDENTS

 CHARACTER, (LEN=20) :: SURNAME

 CHARACTER, (LEN=10) :: NAME

 INTEGER

 :: GRADE

END TYPE STUDENTS

STUDENTS

 :: STUDENT

CONTAINS



Παράδειγμα 3 (...συνέχεια)



! ΔΗΛΩΣΕΙΣ ΣΥΝΑΡΤΗΣΕΩΝ, ΔΙΑΔΙΚΑΣΙΩΝ

```
REAL FUNCTION SUM(X,Y)
```

```
IMPLICIT NONE
```

```
REAL, INTENT(IN)      :: X,Y
```

```
SUM=X+Y
```

```
END FUNCTION SUM
```

```
END MODULE EXAMPLE
```



Με ποιο τρόπο καλούμε το module από το κυρίως πρόγραμμα;

USE «όνομα module»

ή

USE «όνομα module», **ONLY** : όνομα1, όνομα2, ..., όνομα N

Παράδειγμα.

PROGRAM MAIN

USE EXAMPLE, ONLY : PI, G

IMPLICIT NONE

.....

END PROGRAM MAIN



Παράδειγμα 4

Να γραφεί module που να περιέχει συναρτήσεις για την πρόσθεση δυαδικών μεταβλητών, π.χ.

+	0	1
0	0	1
1	1	1



To Module

```
MODULE BOOLEAN_OPERATIONS
```

```
IMPLICIT NONE
```

```
CONTAINS
```

```
    INTEGER FUNCTION BOOL_ADD(X,Y)
```

```
        IMPLICIT NONE
```

```
        INTEGER, INTENT(IN) :: X,Y
```

```
        IF ((X==0).AND.(Y==0)) THEN
```

```
            BOOL_ADD=0
```

```
        ELSE
```

```
            BOOL_ADD=1
```

```
        END IF
```

```
    END FUNCTION BOOL_ADD
```

```
END MODULE BOOLEAN_OPERATIONS
```



Το πρόγραμμα που καλεί το module

```
PROGRAM OPERATIONS
```

```
    USE BOOLEAN_OPERATIONS
```

```
    IMPLICIT NONE
```

```
    INTEGER          :: A,B
```

```
    PRINT*,"A,B="
```

```
    READ*,A,B
```

```
    PRINT*,"A+B=",BOOL_ADD(A,B)
```

```
END PROGRAM OPERATIONS
```



Υπέρβαση τελεστών στο προηγούμενο παράδειγμα

```
MODULE BOOLEAN_OPERATIONS  
  IMPLICIT NONE  
  
  INTERFACE OPERATOR(.ADD.)  
    MODULE PROCEDURE BOOL_ADD  
  END INTERFACE  
  
CONTAINS  
  INTEGER FUNCTION BOOL_ADD(X,Y)  
    .....  
  END FUNCTION BOOL_ADD  
END MODULE BOOLEAN_OPERATIONS
```



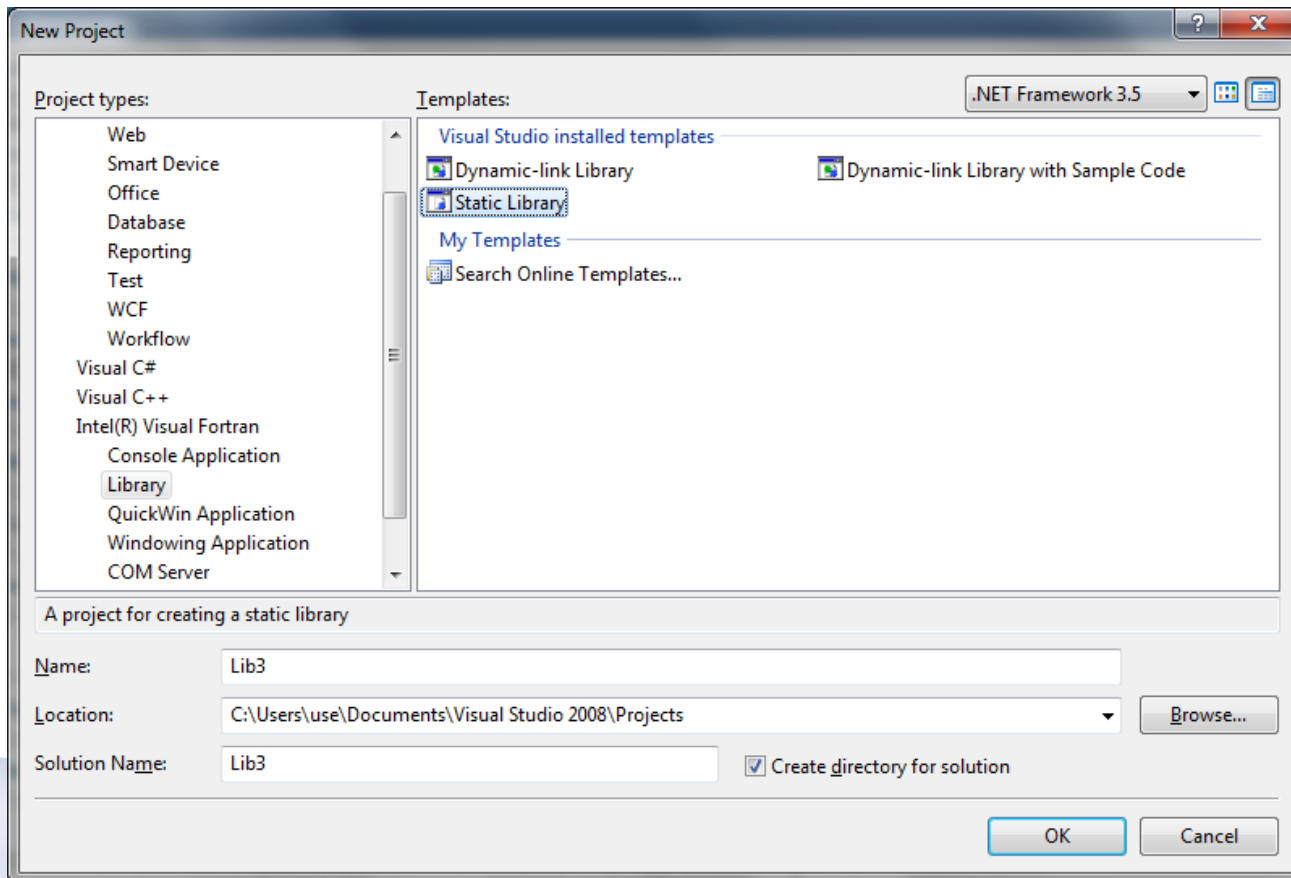
Το πρόγραμμα που καλεί τον νέο τελεστή

```
PROGRAM OPERATIONS
  USE BOOLEAN_OPERATIONS
  .....
  PRINT*,"A+B=",A.ADD.B
  PRINT*,"A*B=",A.MUL.B
END PROGRAM OPERATIONS
```



Δημιουργία στατικής βιβλιοθήκης (Static Library)

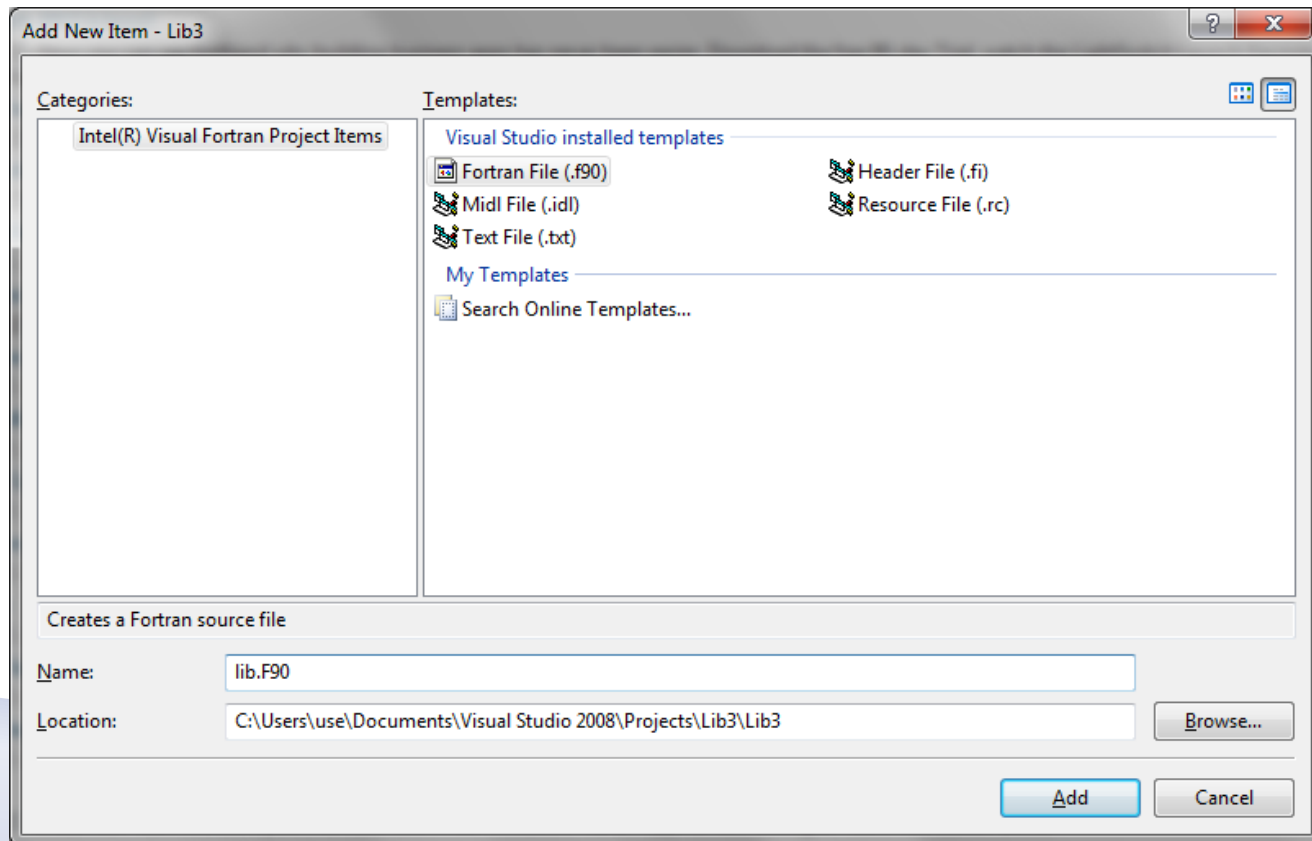
File → New → Project → Library → Static Library



Δημιουργία στατικής βιβλιοθήκης (Πρόσθεση Αρχείου)

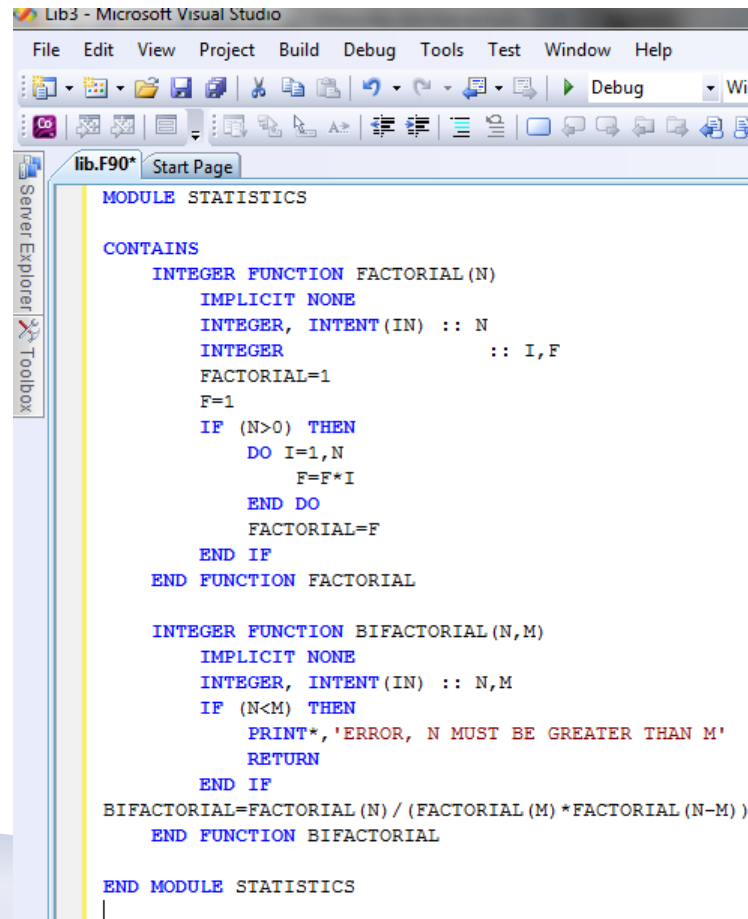
Προσθέτουμε ένα νέο αρχείο

Project → Add New Item (ή Ctrl+Shift+A) → Fortran File (.F90).



Δημιουργία στατικής βιβλιοθήκης (Πρόσθεση module)

Προσθέτουμε το module που δημιουργήσαμε και κάνουμε Build Solution (Ctrl+Shift+B).



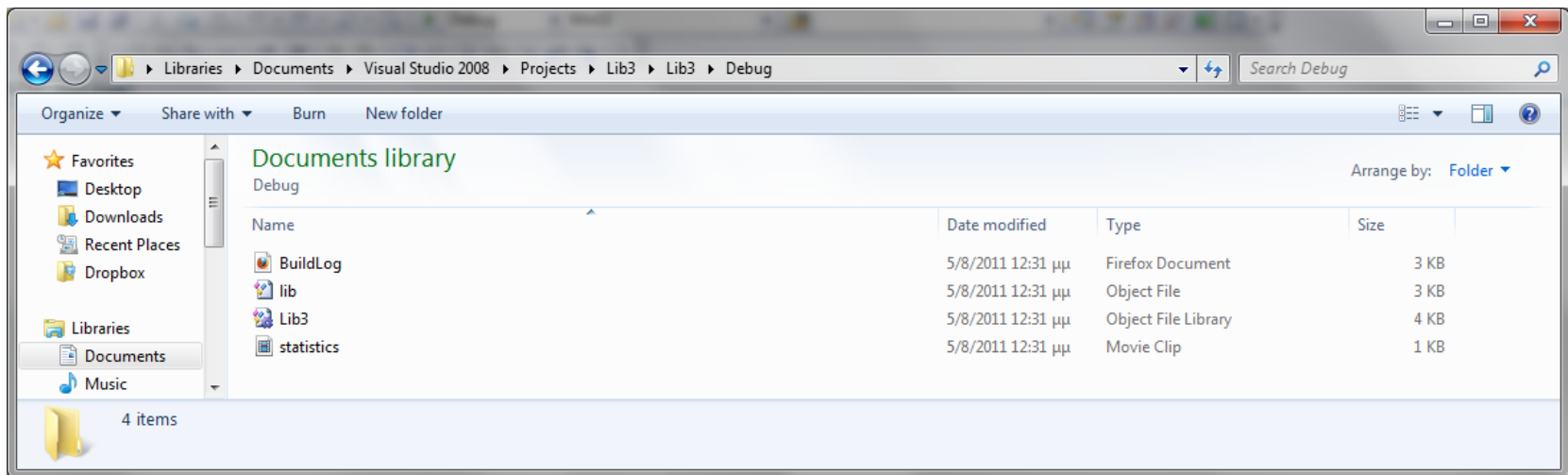
```
MODULE STATISTICS
CONTAINS
  INTEGER FUNCTION FACTORIAL(N)
    IMPLICIT NONE
    INTEGER, INTENT(IN) :: N
    INTEGER                :: I,F
    FACTORIAL=1
    F=1
    IF (N>0) THEN
      DO I=1,N
        F=F*I
      END DO
      FACTORIAL=F
    END IF
  END FUNCTION FACTORIAL

  INTEGER FUNCTION BIFACTORIAL(N,M)
    IMPLICIT NONE
    INTEGER, INTENT(IN) :: N,M
    IF (N<M) THEN
      PRINT*, 'ERROR, N MUST BE GREATER THAN M'
      RETURN
    END IF
    BIFACTORIAL=FACTORIAL(N) / (FACTORIAL(M) *FACTORIAL(N-M))
  END FUNCTION BIFACTORIAL
END MODULE STATISTICS
```



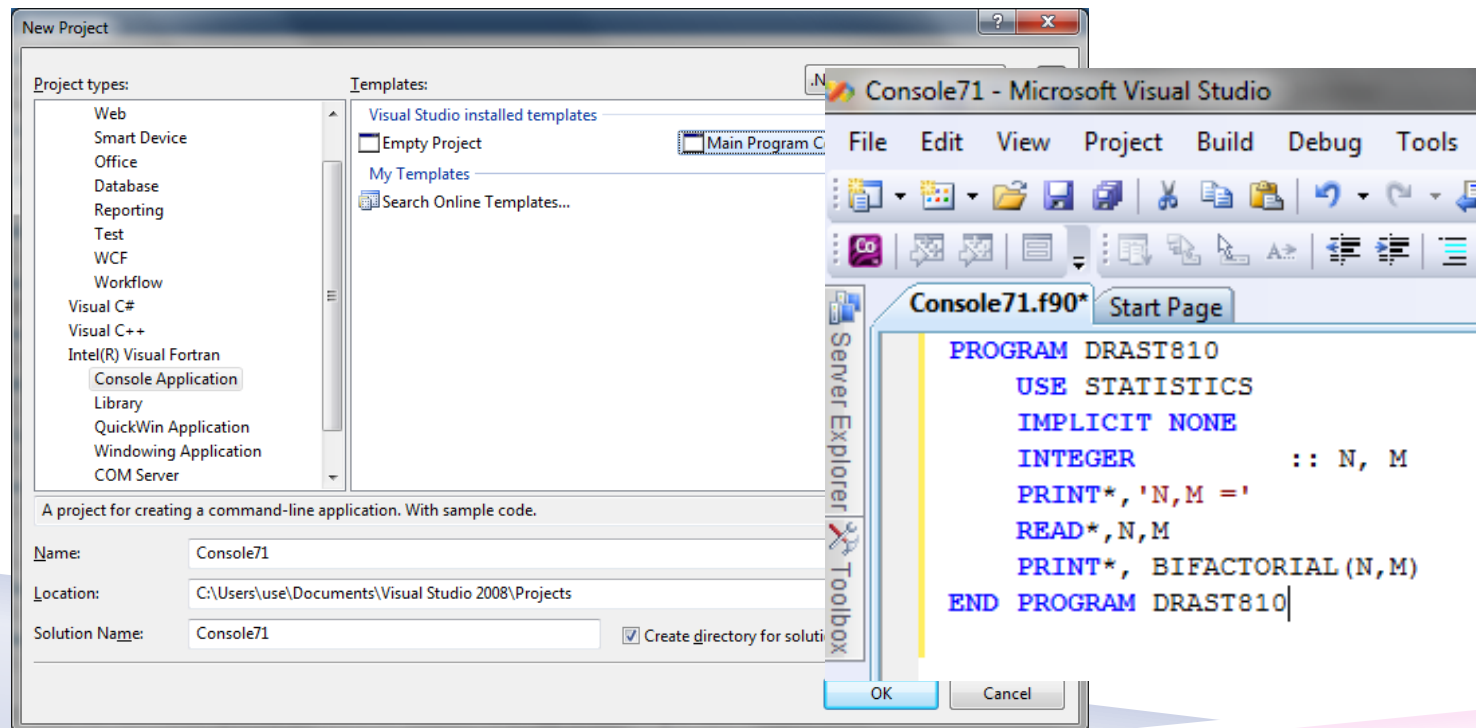
Δημιουργία στατικής βιβλιοθήκης (Build)

Κατά την διάρκεια του Build θα δούμε ότι δημιουργήθηκε, στον φάκελο που υπάρχει το project που δημιουργήσαμε, το αντικείμενο αρχείο lib3.lib



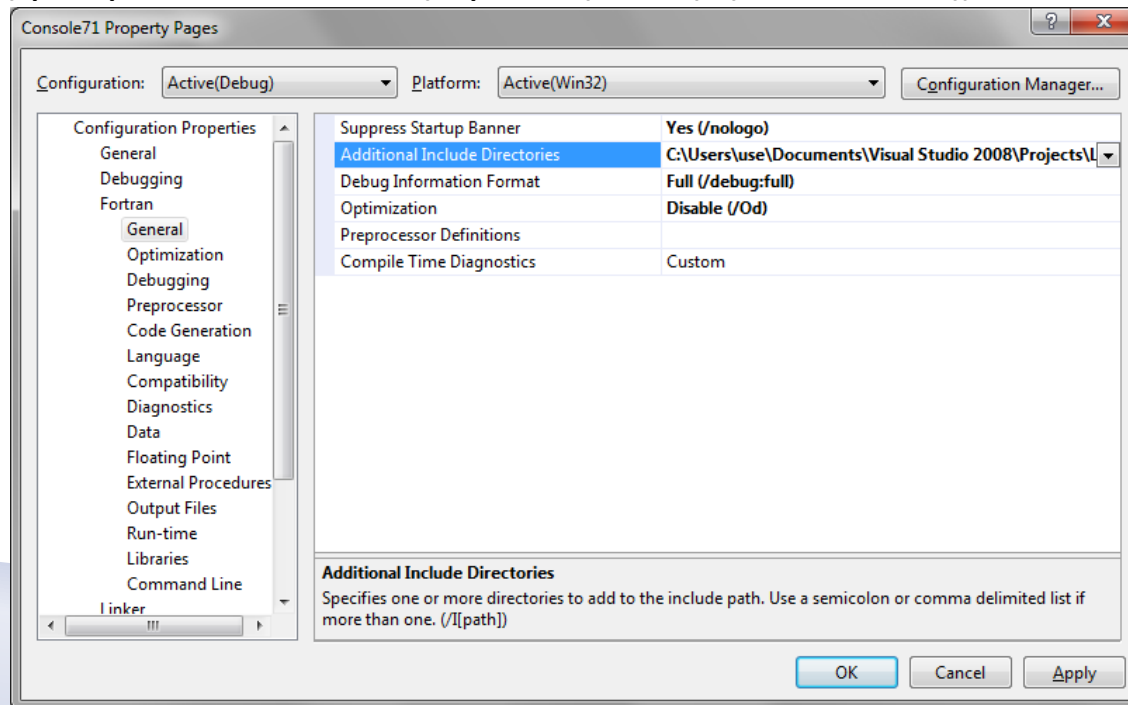
Δημιουργία στατικής βιβλιοθήκης (Χρήση του module)

Κλείνουμε το project που δουλεύαμε (File → Close Solution) και δημιουργούμε ένα νέο project όπου θα γράψουμε το νέο πρόγραμμα (Console Application) που θα καλεί το module της βιβλιοθήκης που δημιουργήσαμε File → New Project.



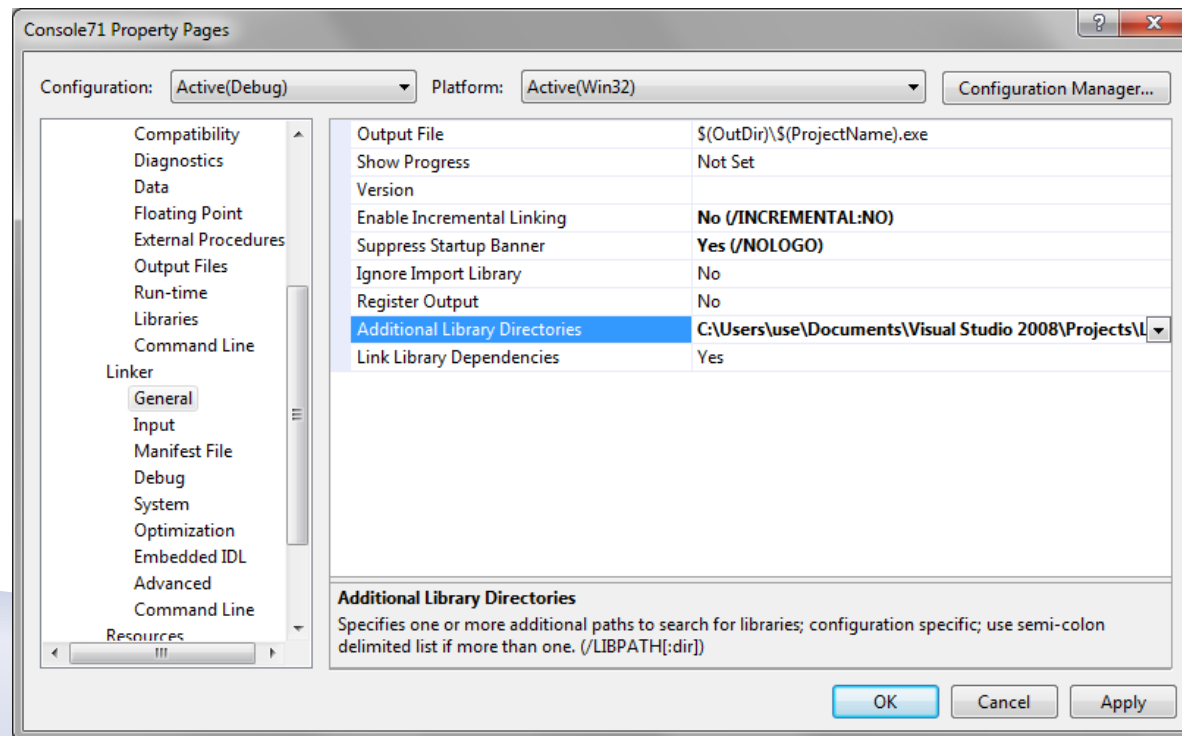
Δημιουργία στατικής βιβλιοθήκης (Εισαγωγή καταλόγου)

Επιλέγουμε από το μενού Project → Properties → Fortran → General και στη συνέχεια προσθέτουμε στην λίστα "Additional Include Directories" τον κατάλογο που βρίσκεται το αρχείο lib3.lib που δημιουργήσαμε π.χ.
C:\Users\use\Documents\Visual Studio 2008\Projects\Lib3\Lib3\Debug (σε άλλο υπολογιστή μπορεί να είναι διαφορετική αυτή η διεύθυνση).



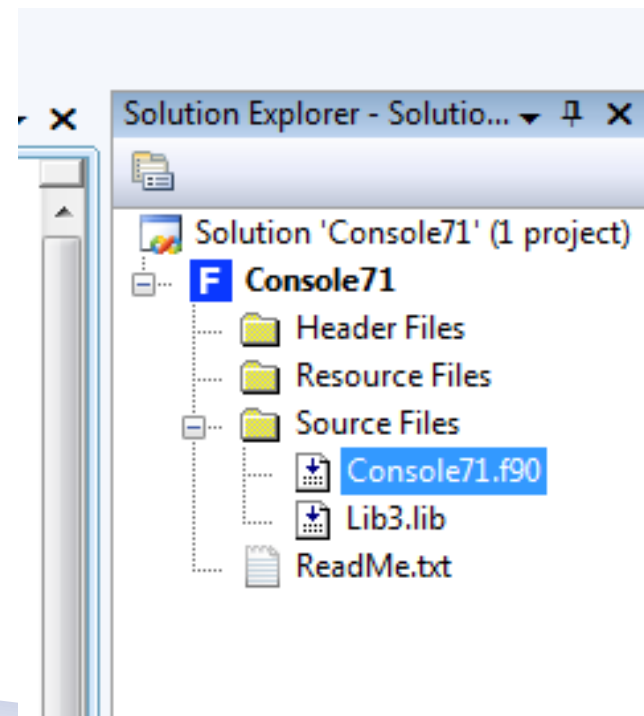
Δημιουργία στατικής βιβλιοθήκης (Επιλογή καταλόγου)

Επιλέγουμε Linker General, και τον κατάλογο που βρίσκεται το αρχείο lib3.lib που δημιουργήσαμε π.χ. C:\Users\use\Documents\Visual Studio 2008\Projects\Lib3\Lib3\Debug (σε άλλο υπολογιστή μπορεί να είναι διαφορετική αυτή η διεύθυνση).



Δημιουργία στατικής βιβλιοθήκης

Προσθέτουμε την βιβλιοθήκη lib3.lib στο project που δημιουργήσαμε (Project Add Existing Item).
Μεταφράζουμε και εκτελούμε το πρόγραμμα.



Σύνδεση βιβλιοθήκης στο αντικείμενο πρόγραμμα

Στην **Compaq Visual Fortran** πριν τη δημιουργία του εκτελέσιμου προγράμματος, κατά την διάρκεια της σύνδεσης, θα πρέπει να συνδέσουμε την βιβλιοθήκη LIB3.LIB στο αντικείμενο πρόγραμμα μας. Θα πρέπει λοιπόν να πάμε στην επιλογή PROJECTS-SETTINGS-LINK και να δηλώσουμε επιπλέον στο Object/library modules την βιβλιοθήκη LIB3.LIB.



Συναρτήσεις – Διαδικασίες της αριθμητικής IMSL βιβλιοθήκης της Compaq Visual Fortran

- α) **MATH/LIBRARY** : διαθέτει συναρτήσεις χρήσιμες σε περιοχές των εφαρμοσμένων μαθηματικών, όπως γραμμικά συστήματα, ανάλυση ιδιοσυστημάτων, προσέγγιση και παρεμβολή, ολοκλήρωση και διαφόριση, διαφορικές εξισώσεις, μετασχηματισμοί, μη γραμμικές εξισώσεις, βελτιστοποίηση, πράξεις πινάκων, καθώς και άλλες εφαρμογές.
- β) **STAT/LIBRARY** : διαθέτει στατιστικές συναρτήσεις που αφορούν βασικά στατιστικά στοιχεία, ανάλυση συσχέτισης, μη παραμετρική στατιστική, συναρτήσεις πιθανότητας, κλπ.



Συναρτήσεις – Διαδικασίες της αριθμητικής IMSL Math Library

IMSL Math Library

- ✓ Γραμμικά συστήματα.
- ✓ Ιδιοτιμές, ιδιοδιανύσματα.
- ✓ Παρεμβολή και μέθοδοι προσέγγισης.
- ✓ Ολοκλήρωση και παραγωγή.
- ✓ Μετασχηματισμοί.
- ✓ Μη γραμμικές εξισώσεις.
- ✓ Βελτιστοποίηση.
- ✓ Βασικές πράξεις πινάκων και διανυσμάτων.
- ✓ Τελεστές γραμμικής άλγεβρας και γενικευμένες συναρτήσεις.



Συναρτήσεις – Διαδικασίες της αριθμητικής

IMSL Math/Library Special Functions

IMSL Math/Library Special Functions

- ✓ Στοιχειώδεις συναρτήσεις.
- ✓ Τριγωνομετρικές και υπερβολικές συναρτήσεις.
- ✓ Εκθετικά ολοκληρώματα και συγγενείς συναρτήσεις.
- ✓ Γάμμα συναρτήσεις και συγγενείς συναρτήσεις.
- ✓ Συναρτήσεις λάθους και συγγενείς συναρτήσεις.
- ✓ Συναρτήσεις Bessel.
- ✓ Συναρτήσεις Kelvin.
- ✓ Συναρτήσεις Airy.
- ✓ Ελλειπτικά ολοκληρώματα.
- ✓ Ελλειπτικές και συγγενείς συναρτήσεις.
- ✓ Συναρτήσεις κατανομής πιθανότητας και συγγενείς συναρτήσεις.
- ✓ Συναρτήσεις Mathieu.



Συναρτήσεις – Διαδικασίες της αριθμητικής IMSL Stat/Library

IMSL Stat/Library

- ✓ Βασικές στατιστικές συναρτήσεις.
- ✓ Παλινδρόμηση.
- ✓ Συσχέτιση.
- ✓ Ανάλυση διασποράς.
- ✓ Διακριτή ανάλυση δεδομένων.
- ✓ Μη παραμετρική στατιστική.
- ✓ Έλεγχος στατιστικών υποθέσεων.
- ✓ Ανάλυση χρονολογικών σειρών και πρόβλεψη.
- ✓ Δομές Συνδιακύμανσης και Παραγοντική Ανάλυση.
- ✓ Cluster Analysis.
- ✓ Πολυδιάστατη Κλιμάκωση.
- ✓ Πυκνότητα και Εκτίμηση Κινδύνου.
- ✓ Συναρτήσεις κατανομής πιθανότητας και αντίστροφες.
- ✓ Γεννήτριες τυχαίων αριθμών.



Παράδειγμα με συναρτήσεις της MATH/IMSL βιβλιοθήκης της Intel Visual Fortran

```
PROGRAM LIBRARY
USE LINRG_INT
USE WRRRN_INT
IMPLICIT NONE
REAL, DIMENSION(:,:), ALLOCATABLE :: A, AINV
INTEGER :: I,J
INTEGER :: N
PRINT*, 'DIMENSION OF THE SQUARE MATRIX ='
READ*, N
ALLOCATE ( A(N,N), AINV(N,N) )
READ*, ((A(I,J),J=1,N),I=1,N)
CALL LINRG(A, AINV)
CALL WRRRN('AINV',AINV)
END PROGRAM LIBRARY
```



Παράδειγμα με συναρτήσεις της MATH/IMSL βιβλιοθήκης της Compaq Visual Fortran

PROGRAM LIBRARY

```
USE NUMERICAL_LIBRARIES
```

```
IMPLICIT NONE
```

```
REAL, DIMENSION(:,:), ALLOCATABLE :: A, AINV
```

```
INTEGER :: I,J
```

```
INTEGER      :: N
```

```
PRINT*,'DIMENSION OF THE SQUARE MATRIX ='
```

```
READ*,N
```

```
ALLOCATE ( A(N,N), AINV(N,N) )
```

```
READ* , ((A(I,J),J=1,N),I=1,N)
```

```
CALL LINRG(N,A,N,AINV,N) ! Υπολογίζει τον αντίστροφο AINV του A
```

```
CALL WRRRN('AINV',N,N,AINV,N,0) ! Εκτυπώνει τον AINV
```

```
END PROGRAM LIBRARY
```



Σύνταξη της LINRG

LINRG (N, A, LDA, AINV, LDAINV)

Ορίσματα

N — Διάσταση του πίνακα A. (Είσοδος)

A — Πίνακας N επί N, του οποίου ζητάμε τον αντίστροφο.
(Είσοδος)

LDA — Η διάσταση του A όπως αυτή ορίστηκε στο πρόγραμμα που καλεί τη διαδικασία. (Είσοδος)

AINV — Ο αντίστροφος πίνακας του A. (Εξοδος)

Εάν ο A δε χρειάζεται μπορούμε να χρησιμοποιήσουμε στην θέση του AINV τον A και συνεπώς θα μοιραστούν την ίδια θέση μνήμης.

LDAINV — Η διάσταση του AINV όπως ακριβώς έχει οριστεί στο πρόγραμμα που καλεί τη διαδικασία. (Είσοδος)



Σύνταξη της WRRRN

WRRRN (TITLE, NRA, NCA, A, LDA, ITRING)

Ορίσματα

- TITLE** — Αλυσίδα χαρακτήρων η οποία προσδιορίζει τον τίτλο που θα εμφανισθεί πριν τον πίνακα που θα εκτυπωθεί. (Είσοδος)
- NRA** — Πλήθος γραμμών. (Είσοδος)
- NCA** — Πλήθος στηλών. (Είσοδος)
- A** — Ο NRA επί NCA πίνακας που πρόκειται να εκτυπωθεί. (Είσοδος)
- LDA** — Η διάσταση του πίνακα A όπως αυτή δηλώθηκε στο πρόγραμμα που καλεί την διαδικασία. (Είσοδος)
- ITRING** — Επιλογή εκτύπωσης. (Είσοδος)



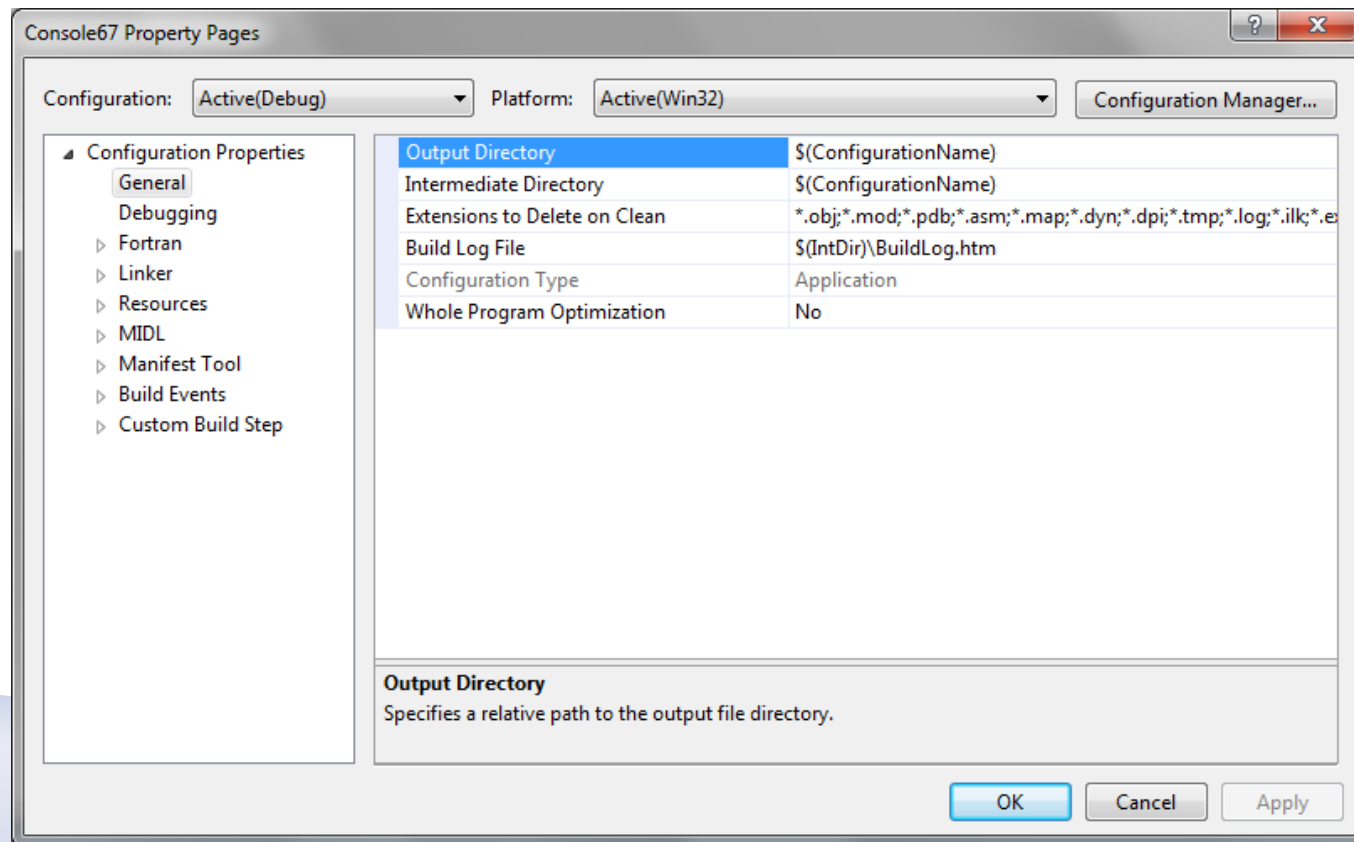
Επιλογή εκτύπωσης ITRING

0	Ο πλήρης πίνακας εκτυπώνεται.
1	Ο πίνακας πάνω από την κύρια διαγώνιο του A εκτυπώνεται, συμπεριλαμβανομένης και της κυρίας διαγωνίου.
2	Ο πίνακας πάνω από την κύρια διαγώνιο του A εκτυπώνεται, μη συμπεριλαμβανομένης της κυρίας διαγωνίου.
-1	Ο πίνακας κάτω από την κύρια διαγώνιο του A εκτυπώνεται, συμπεριλαμβανομένης και της κυρίας διαγωνίου.
-2	Ο πίνακας πάνω από την κύρια διαγώνιο του A εκτυπώνεται, μη συμπεριλαμβανομένης της κυρίας διαγωνίου.



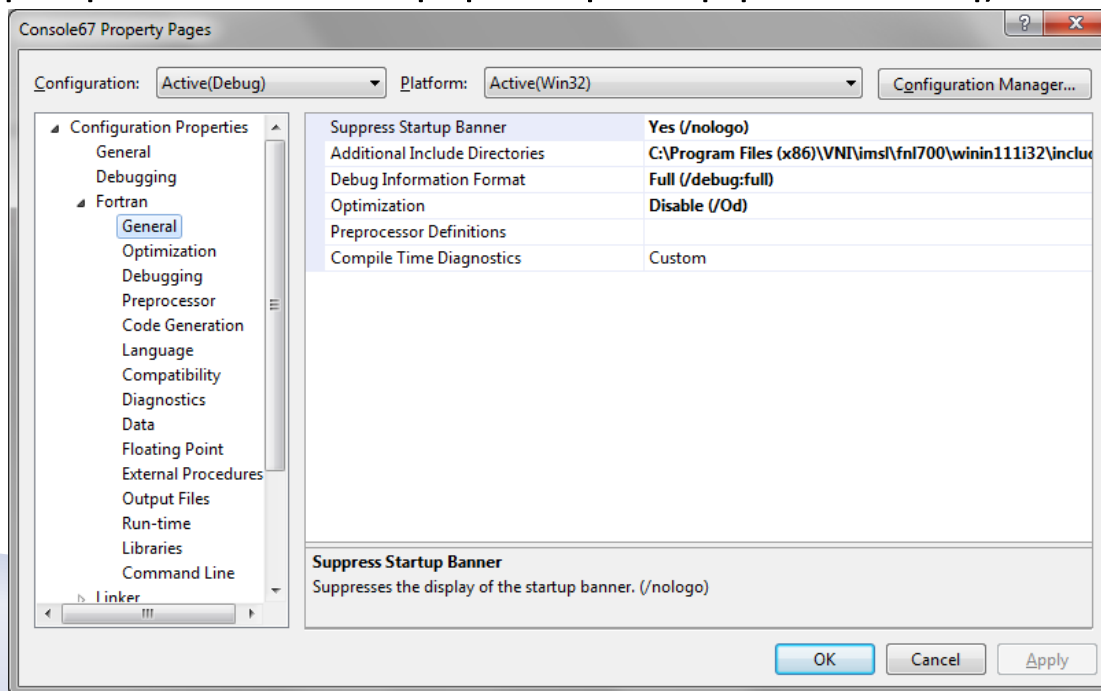
Διαδικασία μετάφρασης/εκτέλεσης προγραμμάτων με συναρτήσεις από την IMSL βιβλιοθήκη (Βήμα 1)

- Επιλέγουμε από το μενού Project → Properties → Configuration Properties.



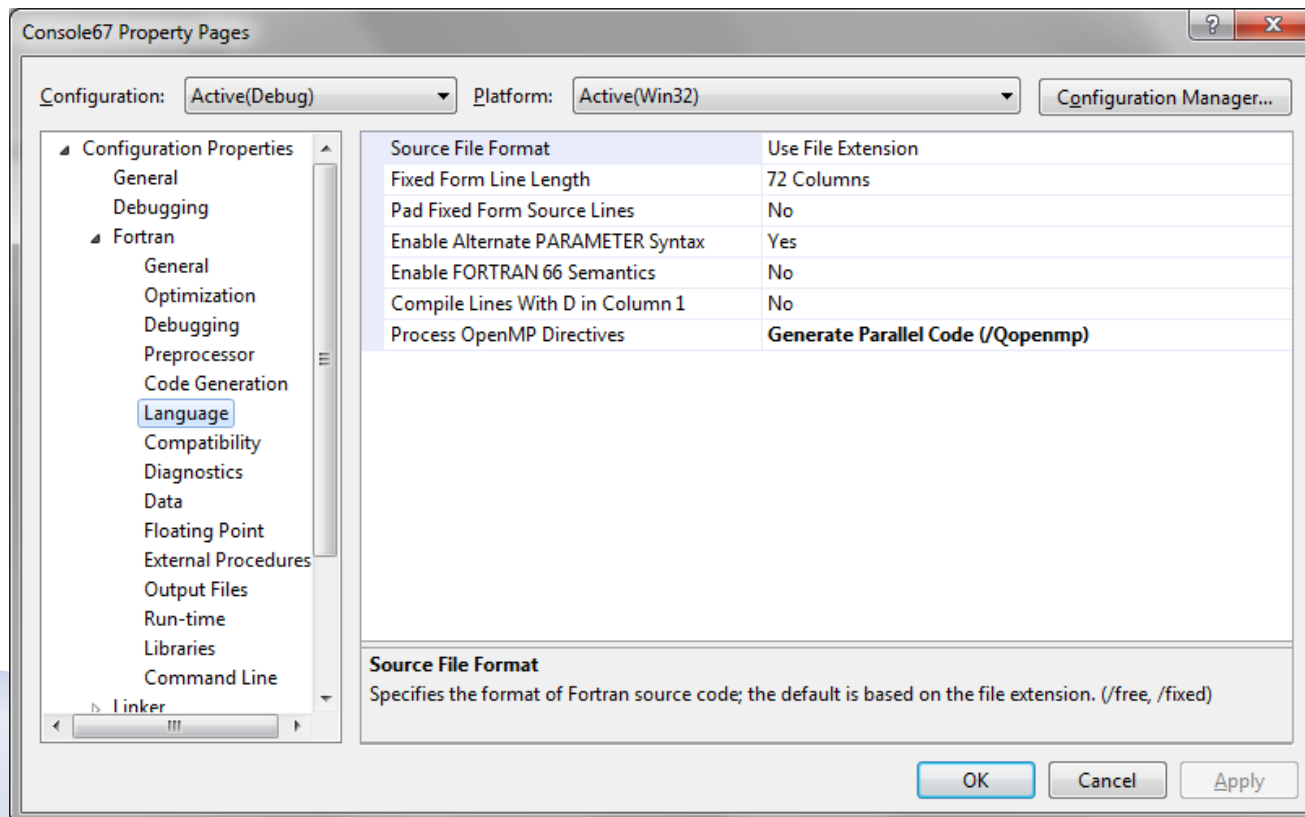
Διαδικασία μετάφρασης/εκτέλεσης προγραμμάτων με συναρτήσεις από την IMSL βιβλιοθήκη (Βήμα 2)

- Κάτω από την επιλογή Fortran επιλέγουμε General και στη συνέχεια προσθέτουμε στην λίστα "Additional Include Directories" την διεύθυνση από τον κατάλογο που βρίσκεται ο DLL φάκελος από την IMSL βιβλιοθήκη στον υπολογιστή μας π.χ. C:\Program Files (x86)\VNI\imsl\fnl700\winin111i32\include\dll (σε άλλο υπολογιστή μπορεί να είναι διαφορετική αυτή η διεύθυνση).



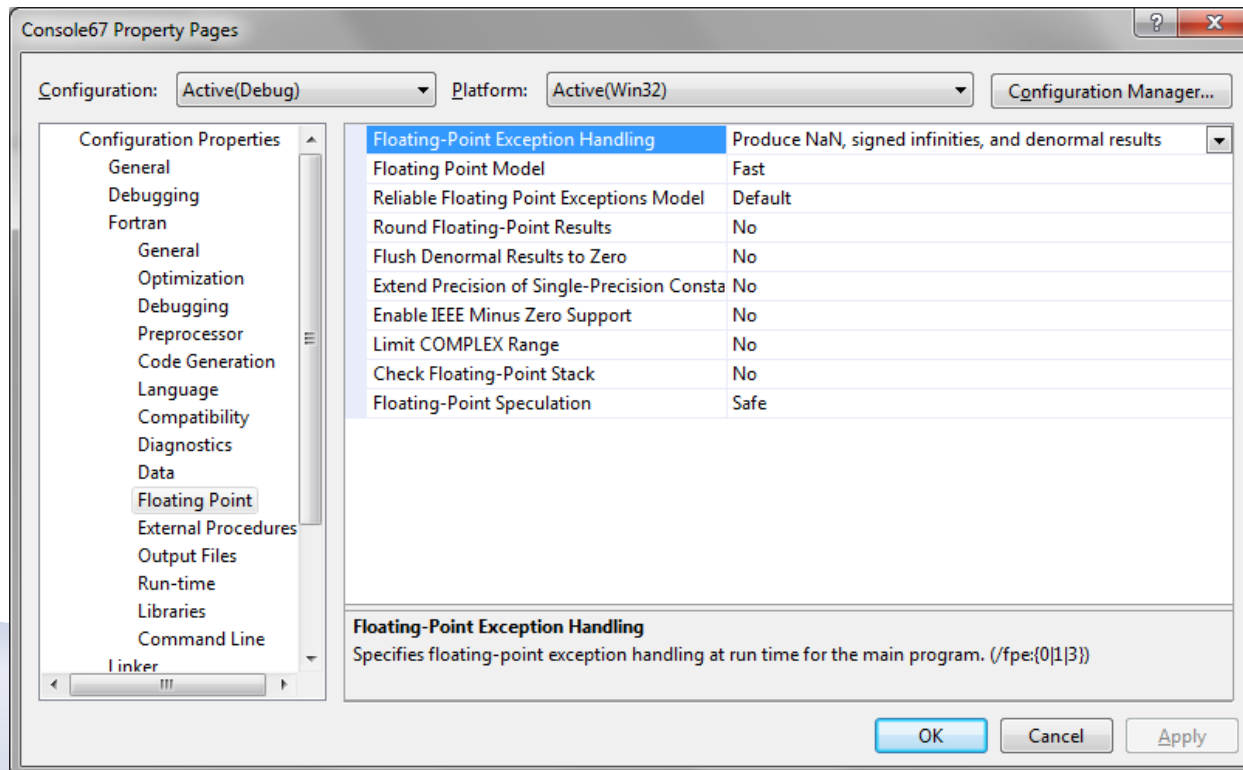
Διαδικασία μετάφρασης/εκτέλεσης προγραμμάτων με συναρτήσεις από την IMSL βιβλιοθήκη (Βήμα 3)

- Επιλέγουμε Fortran → Language και θέτουμε στην επιλογή *Process openMP Directives* την τιμή *Generate Parallel Code (/Qopenmp)*.



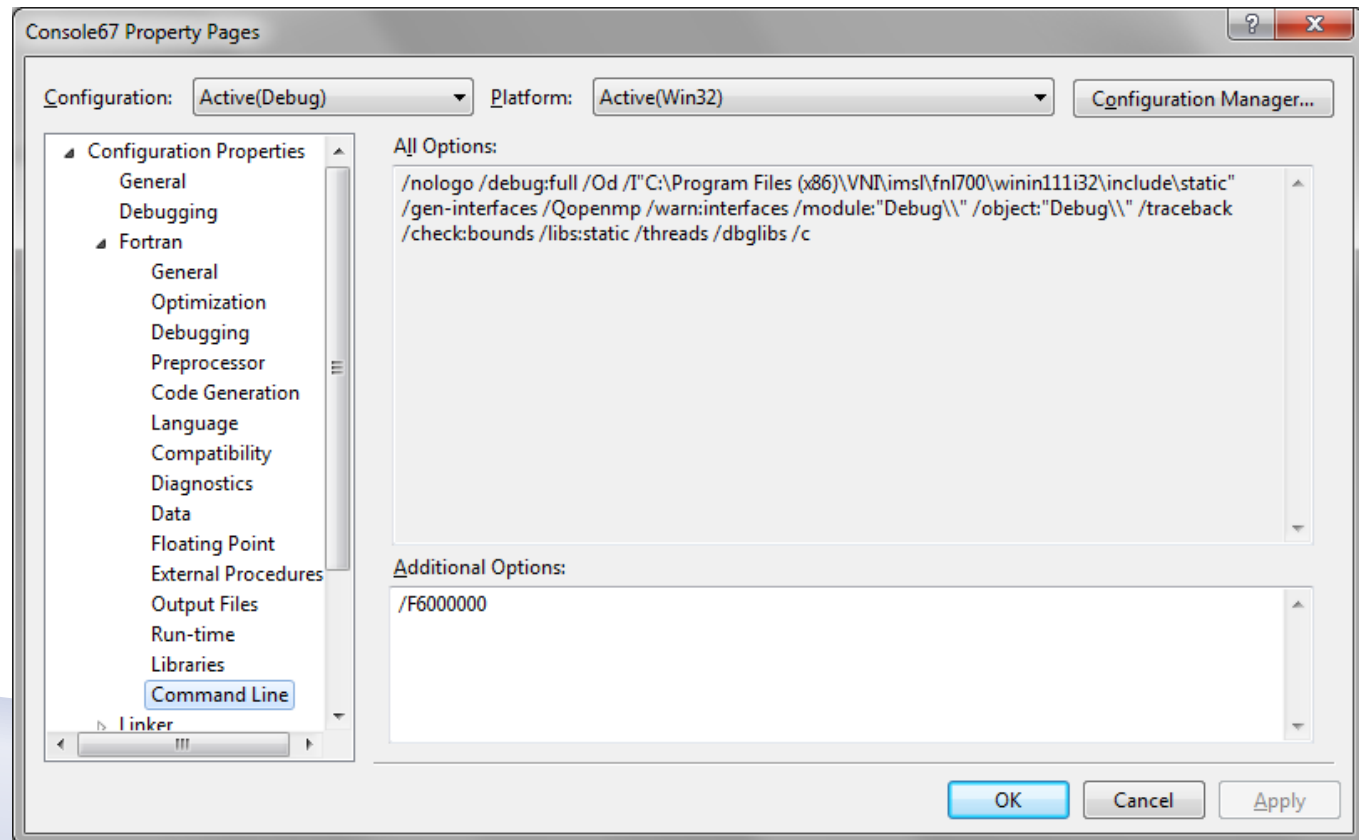
Διαδικασία μετάφρασης/εκτέλεσης προγραμμάτων με συναρτήσεις από την IMSL βιβλιοθήκη (Βήμα 4)

- Επιλέγουμε Fortran → Floating Point, και ελέγχουμε αν η στάνταρ επιλογή για Floating Point Exception Handling έχει ως τιμή "Produce NaN, signed infinity, and denormal results"



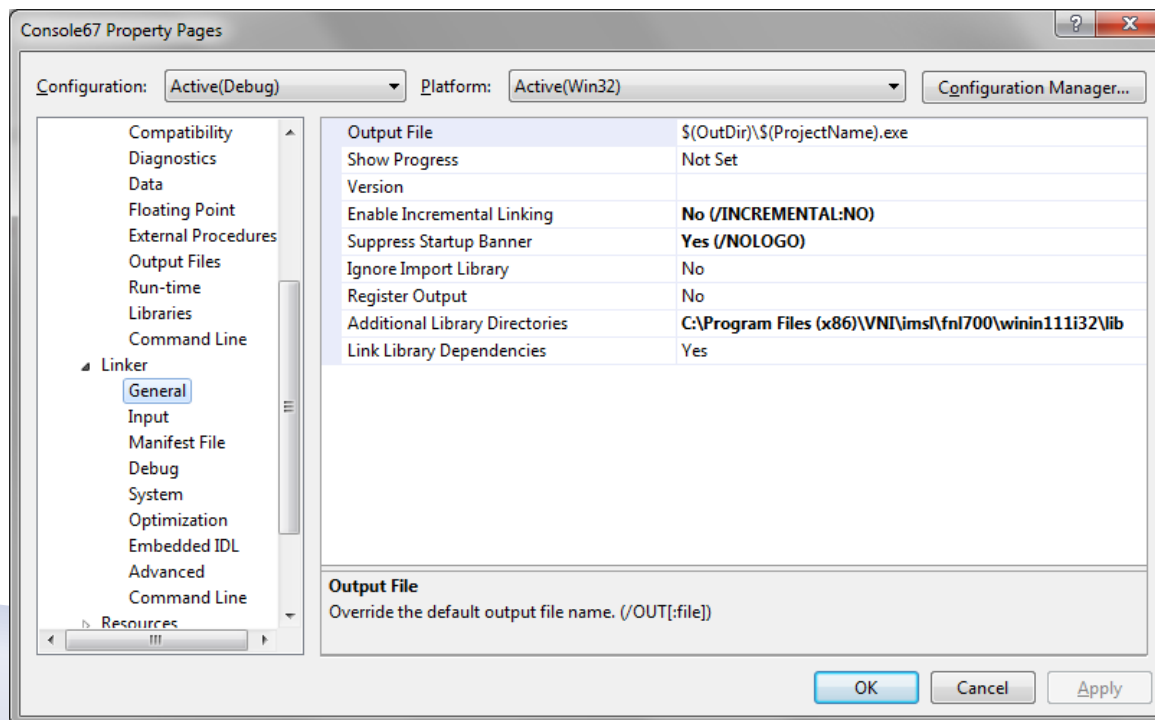
Διαδικασία μετάφρασης/εκτέλεσης προγραμμάτων με συναρτήσεις από την IMSL βιβλιοθήκη (Βήμα 5)

- Επιλέγουμε Fortran → Command Line, και προσθέτουμε στο πεδίο από το Additional Options την τιμή /F6000000.



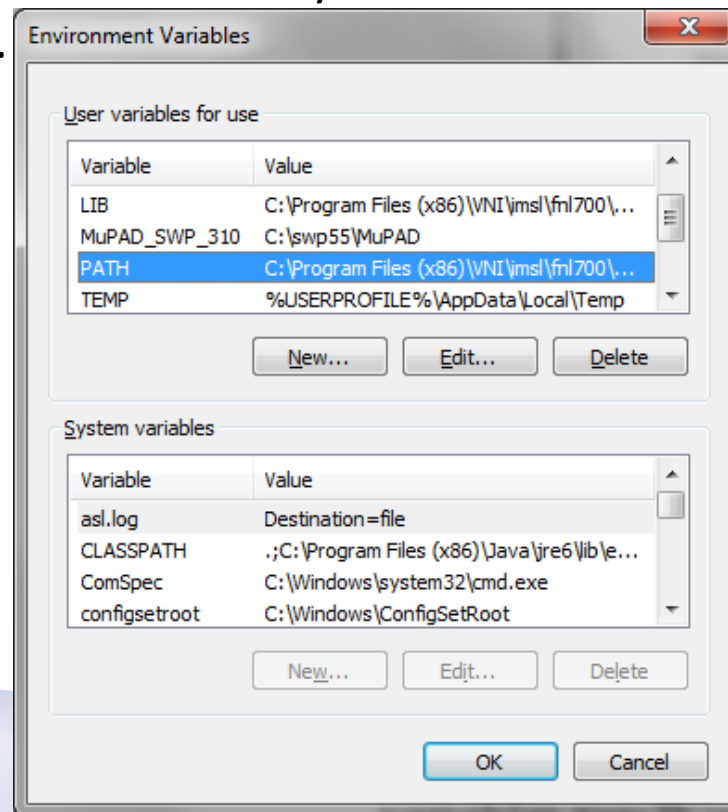
Διαδικασία μετάφρασης/εκτέλεσης προγραμμάτων με συναρτήσεις από την IMSL βιβλιοθήκη (Βήμα 6)

- Επιλέγουμε Linker → General, και προσθέτουμε τον κατάλογο που βρίσκεται η IMSL βιβλιοθήκη στην επιλογή "Additional Library Directories" π.χ. C:\Program Files (x86)\VNI\imsl\fnl700\winin111i32\lib (σε άλλο υπολογιστή μπορεί να είναι διαφορετική αυτή η διεύθυνση).



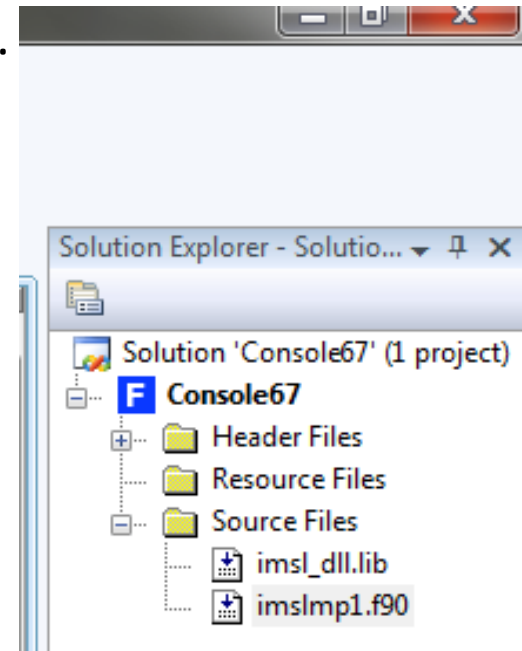
Διαδικασία μετάφρασης/εκτέλεσης προγραμμάτων με συναρτήσεις από την IMSL βιβλιοθήκη (Βήμα 7)

- Ο κατάλογος αυτός πρέπει να προστεθεί και στην επιλογή PATH στα environment variables των Windows πριν να ξεκινήσει το Microsoft Visual Studio (Control Panel → System – Edit environment variables for your account).



Διαδικασία μετάφρασης/εκτέλεσης προγραμμάτων με συναρτήσεις από την IMSL βιβλιοθήκη (Βήμα 8)

- Προσδιορίζουμε ποια από τις IMSL βιβλιοθήκες θα χρειαστούμε να συνδέσουμε στο project που εργαζόμαστε. Στο παραπάνω παράδειγμα προσθέσαμε την βιβλιοθήκη `imsl_dll.lib` στον Project Explorer (Project Add → Existing Item).
- Μεταφράζουμε και εκτελούμε το πρόγραμμα.



Διαδικασία μετάφρασης/εκτέλεσης προγραμμάτων με συναρτήσεις από την IMSL βιβλιοθήκη

Αφού γράψουμε το πρόγραμμά μας, στη συνέχεια κάνουμε τη μετάφραση του προγράμματος. Πριν τη δημιουργία όμως του εκτελέσιμου προγράμματος, κατά την διάρκεια της σύνδεσης, θα πρέπει να συνδέσουμε την βιβλιοθήκη SMATHS.LIB στο αντικείμενο πρόγραμμα μας, αν και αυτό συνήθως γίνεται αυτόματα. Θα πρέπει λοιπόν να πάμε στην επιλογή PROJECTS-SETTINGS-LINK και να δηλώσουμε επιπλέον στο Object/library modules την βιβλιοθήκη SMATHS.LIB, π.χ. KERNEL32.LIB ; SMATHS.LIB, η οποία περιέχει τις παραπάνω συναρτήσεις (αντίστοιχα την STATS.LIB για στατιστικές συναρτήσεις) που χρησιμοποιήσαμε. (Για περισσότερες πληροφορίες δες εγχειρίδιο χρήσης της Compaq Visual Fortran).



Άσκηση 2

Να γραφεί πρόγραμμα το οποίο θα υπολογίζει τις ιδιοτιμές και τα ιδιοανύσματα του πίνακα

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{pmatrix}$$



Βιβλιογραφία

Ν. Καραμπετάκης, *Εισαγωγή στη Fortran 90/95/2003*, 2^η Έκδοση, Θεσσαλονίκη: Εκδόσεις Ζήτη, 2011.



Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Καραμπετάκης Νικόλαος. «Εισαγωγή στον Προγραμματισμό Η/Υ (Fortran 90/95/2003). Διαδικασίες». Έκδοση: 1.0. Θεσσαλονίκη 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: <http://eclass.auth.gr/courses/OCRS145/>.



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-sa/4.0/>





Τέλος Ενότητας

Επεξεργασία: Αναστασία Γ. Γρηγοριάδου
Θεσσαλονίκη, Χειμερινό Εξάμηνο 2013-2014

