

# Egzamin Fortran

## Spis treści

<b>1</b>	<b>Opracowanie zagadnień</b>	<b>3</b>
1.1	Reguły zapisu instrukcji w Fortranie <sup>77</sup>	3
1.2	Postacie stałej rzeczywistej. Podać przykłady	3
1.3	Postać stałej podwójnej precyzji. Podać przykłady	3
1.4	Na czym polega reguła pierwszej litery	3
1.5	Podać przykład zastosowania dyrektywy IMPLICIT	3
1.6	Co musi wystąpić po dyrektywie IMPLICIT NONE?	3
1.7	Sposoby deklaracji wymiaru i rozmiaru tablicy (zmiennej indeksowanej). Jaki jest maksymalny wymiar tablicy?	4
1.8	Jaka jest różnica między funkcjami wewnętrznymi ATAN i ATAN2?	4
1.9	Wymień operatory arytmetyczne i kolejność ich wykonywania	4
1.10	Wymień operatory relacji	4
1.11	Wymień operatory logiczne	4
1.12	Podaj typ wyniku i jego wartość: 2/4, 2./4, 2d0/4, 5/2, 2./5	5
1.13	Podaj postać bezwarunkowej instrukcji skoku	5
1.14	Podaj postać instrukcji warunkowej prostej	5
1.15	Podaj postać blokowej instrukcji warunkowej	5
1.16	Podaj postać instrukcji warunkowej złożonej	5
1.17	Podaj postać arytmetycznej instrukcji warunkowej	6
1.18	Podaj postać instrukcji cyklu	6
1.19	Co to jest urządzenie standardowe? Podaj postać instrukcji czytania danych z urządzenia standardowego	6
1.20	Podaj postać instrukcji pisania wartości tablicy jednowymiarowej z wykorzystaniem listy cyklu (DO implikowanego)	6
1.21	Podaj postać arytmetycznej instrukcji warunkowej	6
1.22	W jaki sposób przekazywane są parametry wejściowe do segmentu function	7
1.23	Ile wartości może być wyznaczonych w segmencie function i jak są zwracane do modułu nadrzędnego	7
1.24	W jaki sposób przekazywane są parametry wejściowe do procedury subroutine	7
1.25	Ile wartości może być wyznaczonych w procedurze subroutine i jak są zwracane do modułu nadrzędnego	8
1.26	W jaki sposób wartości zmiennych określone w jednym segmencie mogą być dostępne w innym module	8
1.27	Podaj postać deklaracji COMMON	8
1.28	W różnych segmentach może wystąpić deklaracja COMMON o tej samej nazwie. Co można powiedzieć o zmiennych wyszczególnionych w tych deklaracjach	8
1.29	Jaki jest cel stosowania segmentu BLOCK DATA	9
1.30	Podać postać i cel stosowania instrukcji INCLUDE	9

# 1 Opracowanie zagadnień

## 1.1 Reguły zapisu instrukcji w Fortranie77

- Kolumna 1 : znak C, c lub \* oznacza linię komentarza i nie mają wpływu na wykonanie programu. Komentarze można umieszczać także po 72 kolumnie lub na prawo od znaku!
- Kolumny 1-5 : etykieta (ciąg maksymalnie pięciu cyfr, co najmniej jedna niezerowa; umożliwia odwołanie się do etykietowanej linii w programie)
- Kolumna 6 : dowolny znak (różny od zera i spacji) oznacza kontynuację poprzedniej linii. jedna instrukcja może składać się maksymalnie z 20. linii (wierszy)
- Kolumny 7-72 : instrukcje FORTRANu

## 1.2 Postacie stałej rzeczywistej. Podać przykłady

```
1 REAL :: x !12.0, -.3, 1.35E-1
```

Listing 1: dyrektywa implicit

## 1.3 Postać stałej podwójnej precyzji. Podać przykłady

```
1 double precision :: foo !3.54D0, 35.4D-1
```

Listing 2: dyrektywa implicit

## 1.4 Na czym polega reguła pierwszej litery

Jeśli zmienna nie zostanie zadeklarowana to Fortran77 przyjmie regułę pierwszej litery w nazwie.

- Zmienna o nazwie zaczynające się od i, j, k, l, m, n zostaje automatycznie przypisana do typu INTEGER
- pozostałe do typu REAL.

## 1.5 Podać przykład zastosowania dyrektywy IMPLICIT

```
1 program test
2   implicit none
3   integer :: a, b, c
4   ...
5 end program
```

Listing 3: dyrektywa implicit

## 1.6 Co musi wystąpić po dyrektywie IMPLICIT NONE?

Deklaracja stałych (anuluje regułę pierwszej litery (1.4))

### 1.7 Sposoby deklaracji wymiaru i rozmiaru tablicy (zmiennej indeksowanej). Jaki jest maksymalny wymiar tablicy?

Maksymalny 7-wymiarowa tablica.

- **TYP** <nazwa> **DIMENSION** <nazwa>(n1:m1,n2:m2)
- **TYP** <nazwa>(n1:m1,n2:m2)

### 1.8 Jaka jest różnica między funkcjami wewnętrznymi ATAN i ATAN2?

- **ATAN(x)** - arctg w radianach
- **ATAN2(x,y)** - x,y-wspł. wektora, wynik w radianach

### 1.9 Wymień operatory arytmetyczne i kolejność ich wykonywania

Zgodnie z priorytetem(jeśli równoważne to od prawej strony):

- potęgowanie  $A^{**}B$
- mnożenie  $A*B$ , dzielenie  $A/B$
- dodawanie, odejmowanie

### 1.10 Wymień operatory relacji

- **.LT.**
- **.LE.**
- **.EQ.**
- **.NE.**
- **.GE.**
- **.GT.**

### 1.11 Wymień operatory logiczne

- **.NOT.**
- **.AND.**
- **.OR.**
- **.EQV.**

- .EQV. równoważność
- .NEQV.

a	b	.NOT.a	a.AND.b	a.OR.b	a.EQV.b	a.NEQV.b
T	T	F	T	T	T	F
T	F	F	F	T	F	T
F	T	T	F	T	F	T
F	F	T	F	F	T	F

### 1.12 Podaj typ wyniku i jego wartość: 2/4, 2./4, 2d0/4, 5/2, 2./5

2/4=0, 2./4=0.500000000, d20/4=0.500000000000000000, 5/2=2, 2.5=0.400000006

### 1.13 Podaj postać bezwarunkowej instrukcji skoku

GO TO <etykieta>

```
1 10  n = n + 1
2  go to 10
```

Listing 4: cos

### 1.14 Podaj postać instrukcji warunkowej prostej

IF (wyrażenie logiczne) instrukcja

```
1  if (foo.LE.2) bar=2
```

Listing 5: cos

### 1.15 Podaj postać blokowej instrukcji warunkowej

```
1 IF ( wyrażenie logiczne ) THEN
2 ...
3 ...
4 END IF
```

Listing 6: cos

### 1.16 Podaj postać instrukcji warunkowej złożonej

```
1 IF ( wyrażenie logiczne ) THEN
2 ...
3 ELSE IF (warunek) THEN
4 ...
5 ELSE
6 ...
```

```
7 END IF
```

Listing 7: cos

## 1.17 Podaj postać arytmetycznej instrukcji warunkowej

Listing 8: cos

## 1.18 Podaj postać instrukcji cyklu

```
1 DO iterator=start , stop , step
2 ...
3 ...
4 END DO
```

Listing 9: cos

## 1.19 Co to jest urządzenie standardowe? Podaj postać instrukcji czytania danych z urządzenia standardowego

urządzenie wejścia-wyjścia umożliwiające komunikację między programem a środowiskiem zewnętrznym(dysk, ekran ...)

```
1 open(10, file="cache.txt", status="old")
2 read(10,*) foo
3 close(10)
```

Listing 10: cos

## 1.20 Podaj postać instrukcji pisania wartości tablicy jednowymiarowej z wykorzystaniem listy cyklu (DO implikowanego)

```
1 program X
2 dimension T(3)
3 do j=1,3
4   T(j) = float(j**2)
5   write(*,*) T(j)
6   do i=1,2
7
8 end do
9 end program
```

Listing 11: cos

## 1.21 Podaj postać arytmetycznej instrukcji warunkowej

IF(wyrażenie arytmetyczne) etyk1, etyk2, etyk3

Przekierowanie obliczeń do instrukcji oznaczonej odpowiednią etykietą następuje, gdy:

wyrażenie arytmetyczne<0 -> etyk1

wyrażenie arytmetyczne=0 -> etyk2

wyrażenie arytmetyczne>0 -> etyk3

przykład:

```
if(delta) 10,20,30
10 print *,'brak rozwiązań'
go to 100
20 print*, 'jedno rozwiązanie'
go to 100
30 print *, 'dwa rozwiązania'
go to 100
100 continue
```

## 1.22 W jaki sposób przekazywane są parametry wejściowe do segmentu function

-podprogram może być wykonywany z danej jednostki programowej wielokrotnie z różnym zestawem danych.

-funkcja jest wywoływana poprzez podanie jej nazwy wraz z listą parametrów aktualnych ujętych w nawiasy okrągłe.

postać:

zmienna=nazwa(lista parametrów aktualnych)

## 1.23 Ile wartości może być wyznaczonych w segmencie function i jak są zwracane do modułu nadrzędnego

-w segmencie function może być wyznaczony jeden element.

postać:

typ function nazwa (lista parametrów formalnych)

deklaracje

część wykonawcza

nazwa=zwracana wartość

return

end

-instrukcja return powoduje zakończenie wykonywania programu i przekazania sterowania do segmentu z, którego następuje jej wywołanie.

## 1.24 W jaki sposób przekazywane są parametry wejściowe do procedury subroutine

Procedura wywoływana jest w następujący sposób:

call nazwa(lista parametrów aktualnych)

gdzie parametry aktualne to parametry wejściowe.

### 1.25 Ile wartości może być wyznaczonych w procedurze subroutine i jak są zwracane do modułu nadrzędnego

-procedura pozwala na zwracanie większej liczby wartości niż jedna.

-procedura nie ma określonego typu.

ogólna postać:

subroutine nazwa (lista parametrów formalnych)

deklaracje

część wykonawcza

return

end

instrukcja return powoduje zakończenie wykonywania programu i przekazania sterowania do segmentu z, którego następuje jej wywołanie.

### 1.26 W jaki sposób wartości zmiennych określone w jednym segmencie mogą być dostępne w innym module

-takie zmienne można uwspólnić poprzez umieszczenie ich na liście obszarów wspólnych.

-obszar ten musi pojawić się w części deklaracyjnej segmentów, pomiędzy którymi są uwspólnione umieszczane w nim zmienne.

### 1.27 Podaj postać deklaracji COMMON

common /nazwa/ zmienna1, zmienna2

-zmienne mogą być proste lub tablicowe.

jeden obszar może być bez nazwy.

nazwy zmiennych mogą być inne taka sama musi być nazwa obszaru wspólnego i jego długość.

### 1.28 W różnych segmentach może wystąpić deklaracja COMMON o tej samej nazwie. Co można powiedzieć o zmiennych wyszczególnionych w tych deklaracjach

- Jeżeli nazwy zmiennych są inne ale nazwa obszaru wspólnego i jego długość jest taka sama, to zmienne o tej samej liczbie porządkowej są sobie równoważne.



### 1.29 Jaki jest cel stosowania segmentu BLOCK DATA

-Poprzez segment block data mogą być wprowadzane dane do programu.

-segment ten służy do nadawania wartości początkowych zmiennym umieszczonym w obszarach wspólnych.

struktura:

BLOCK DATA nazwa

common /nazwa obszaru/ X,Y,I(10)

data x,y,I /0.0,5.92,4\*3,6\*0/

end

### 1.30 Podać postać i cel stosowania instrukcji INCLUDE

postać: include 'nazwa pliku'

cel: służy do dołączenia do pliku fortranowskiego innego pliku zawierającego procedury, funkcję czy bloki danych.

## Spis rysunków

## Kod źródłowy

1	dyrektywa implicit . . . . .	3
2	dyrektywa implicit . . . . .	3
3	dyrektywa implicit . . . . .	3
4	cos . . . . .	5
5	cos . . . . .	5
6	cos . . . . .	5
7	cos . . . . .	5
8	cos . . . . .	6
9	cos . . . . .	6
10	cos . . . . .	6
11	cos . . . . .	6