

## Grupa 2

### Zadanie 1.

Metoda zstępująca programowania strukturalnego polega na

- A. opracowaniu ogólnego planu rozwiązania problemu, który w każdym kolejnym etapie jest uściślany.
- B. tworzeniu prostych procedur bibliotecznych.
- C. zdefiniowaniu najprostszych procedur i stworzeniu na ich podstawie ogólnego rozwiązania.
- D. programowaniu „od szczegółu do ogółu”.

### Zadanie 2.

Która deklaracja podprogramu **nie pozwoli zwrócić** (przekazać wyniku) **sumy** elementów wektora **A** typu **Twekt**?

- A. `procedure sum (A: Twekt; suma: real);`
- B. `procedure sum (A: Twekt; var suma: real);`
- C. `function suma (A: Twekt): real;`
- D. `procedure sum (var A: Twekt; var suma: real);`

### Zadanie 3.

Która z wymienionych procedur Turbo Pascala tworzy nowy plik „zmienną-plikową” i udostępnia go do zapisu w nim danych?

- A. Reset.
- B. Assign.
- C. Append.
- D. Rewrite.

### Zadanie 4.

Który nagłówek funkcji **jest prawidłowy**?:

- A. `function wynik(A,B:Tab; nazwa:string);`
- B. `function wynik(A,B:Tab; nazwa:string): integer;`
- C. `function wynik(A,B:Tab, nazwa:string): integer;`
- D. `function wynik(A,B:Tab, nazwa:string);`

### Zadanie 5.

Deklaracja tablicy **Arr** typu **Real** o wymiarach **3x3** ma postać

- A. `Arr array[3,3] : Real;`
- B. `Arr : array[1..3,1..3] of Real;`
- C. `Arr array[3,3] of Real;`
- D. `Real arr[3,3];`

### Zadanie 6.

Deklaracja `Function Fun:Real;`

- A. jest nieprawidłowa
- B. deklaruje funkcję zwracającą typ **Real**
- C. deklaruje funkcję o danych wej. typu **Real**
- D. deklaruje funkcję o danych wej. dowolnego typu

### Zadanie 7.

W Pascalu można zadeklarować tablice

- A. tylko jednowymiarowe
- B. tylko jednowymiarowe i dwuwymiarowe
- C. tylko typu **Real**
- D. o dowolnej ilości wymiarów

### Zadanie 8.

Zmienne deklarowane na początku programu, dostępne w każdej procedurze i funkcji, to:

- A. zmienne lokalne
- B. parametry formalne
- C. parametry aktualne
- D. zmienne globalne

### Zadanie 9.

Każda procedura napisana w programie

- A. będzie wywołana tylko raz
- B. Będzie wywołana co najmniej raz
- C, Będzie wywołana zawsze dwukrotnie
- D. Może być wywołana

### Zadanie 10.

Wynikiem wykonania funkcji xxx będzie:

- A. Określenie długości tekstu
- B. Wyświetlenie na ekranie długości tekstu
- C. Wyświetlenie na ekranie wartości zmiennej **tekst**
- D. Odwrócenie wartości zmiennej **tekst** (od ostatniego znaku do pierwszego)

```
function xxx (tekst: string): string;  
var i: byte; s: string;  
begin  
  s := " ";  
  for i := Length (tekst) downto 1 do  
    s:= s+tekst[i];  
  xxx :=s  
end;
```

### Zadanie 11.

Funkcja plikowa, która przyjmuje wartość TRUE jeśli osiągnięty został koniec pliku to:

- A. EOP
- B. EOL
- C. EOF
- D. EOD

### Zadanie 12.

Dla zadeklarowanej obok tablicy w języku Pascal poprawnym zapisem jest

- A. Tab[1] :=a
- B. Tab[a] :=1
- C. Tab[ '1' ] :=a
- D. Tab[ 'a' ] :=1

```
var Tab: array['a'..'z'] of integer;
```

### Zadanie 13.

Blok definicji procedur i funkcji znajduje się:

- A. przed słowem kluczowym Program
- B. po programie głównym
- C. przed programem głównym
- D. w dowolnym miejscu

### Zadanie 14.

Record jest to

- A. Tablica o różnych typach składowych
- B. Słowo kluczowe umożliwiające definiowanie struktur danych
- C. Funkcja zapisu na dysk
- D. Procedura zapisu do pamięci

**Zadanie 15.**

Procedura Append:

- A. otwiera plik do zapisu
- B. otwiera plik w trybie dopisywania danych
- C. otwiera plik do odczytu
- D. nie ma takiej procedury

**Zadanie 16.**

Tablica WYNIKI jest zadeklarowana w następujący sposób:

```
var WYNIKI : array['A'..'F', 1..3] of byte;
```

Ilość elementów, które można zapisać w tablicy WYNIKI wynosi:

- A. 3
- B. 15
- C. 18
- D. 45

**Zadanie 17.**

Przedstawiona obok procedura rekurencyjna `druk(n:integer)`, wywołana z parametrem 5, spowoduje wpisanie na ekranie komputera

- A. liczby 0.
- B. liczby 5.
- C. ciągu liczb 1,2,3,4,5.
- D. ciągu liczb 5,4,3,2,1.

```
procedure druk(n:integer);  
begin  
    if n=0 then  
        exit;  
    writeln(n);  
    druk(n-1);  
end;
```

**Zadanie 18.**

W kodzie źródłowym programu zastosowano procedurę z parametrami formalnymi.

```
Procedure Zadanie (a, b, c: Byte; Var x: Word);
```

W momencie wywołania procedury następuje przekazywanie parametrów przez

- A. wartość.
- B. zmienną.
- C. wartość i zmienną.
- D. parametr zaktualizowany

**Zadanie 19.**

W języku Pascal różnica między funkcją a procedurą polega na tym, że

- A. funkcja zwraca wartość, a procedura nie.
- B. do funkcji jako parametr można przekazywać zmienną, a do procedur tylko wartość.
- C. do procedury jako parametr można przekazywać zmienną, a do funkcji tylko wartość.
- D. funkcja może być zdefiniowana z tylko jednym parametrem, a procedura z wieloma.

**Zadanie 20.**

Indeksowanie tablic w Pascalu zaczyna się od

- A. zera.
- B. końca.
- C. dowolnej wartości całkowitej.
- D. jedynki.

**Zadanie 21.**

Wywołanie procedury `Test (X, Y)` polega na

- A. zapisaniu w zmiennych `X` i `Y` ich zamienionych wzajemnie wartości.
- B. znalezieniu mniejszej z liczb i zapisaniu jej w zmiennej `Pom`
- C. znalezieniu większej z liczb i zapisaniu jej w zmiennej `X`.
- D. zapisaniu w zmiennych `X` i `Y` ich podwojonych wartości.

```
procedure TEST(var A, B: integer);  
var Pom : integer;  
begin  
    Pom:=A;  
    A:=B;  
    B:=Pom;  
end;
```

**Zadanie 22.**

Za pomocą jakiej funkcji kojarzymy zmienną z plikiem:

- A. Rewrite
- B. Assign
- C. Reset
- D. Append

**Zadanie 23.**

Zadeklarowano

```
T : array[1..5] of Real; I:Real; I:=2;
```

Odwołanie do `T[I]` spowoduje

- A. możliwość dostępu do drugiego elementu tablicy `T`
- B. przypisanie elementom tablicy wartości 2
- C. przerwanie programu
- D. błąd kompilacji

**Zadanie 24.**

Przedstawiona procedura rekurencyjna `proc (liczba : integer)` wywołana z parametrem 3, spowoduje wypisanie na ekranie komputera

- A. liczby 4
- B. ciągu liczb 3, 4
- C. ciągu liczb 3, 4, 5
- D. ciągu liczb 4, 5

```
procedure proc (liczba : integer);  
begin  
    liczba:= liczba+1;  
    writeln(liczba);  
    if liczba<5 then  
        proc (liczba);  
    end;
```

**Zadanie 25.**

Przedstawiony fragment programu napisanego w języku PASCAL jest przykładem zastosowania

- A. funkcji iteracyjnej.
- B. funkcji sortowania.
- C. funkcji warunkowej.
- D. funkcji rekurencyjnej.

```
function silnia(n:word) :longint;  
begin  
    if n=0 then  
        silnia:=1  
    else  
        silnia:=silnia(n-1)*n;  
    end;
```

**Zadanie 26.**

Która z poniższych deklaracji typu 50-cio elementowej tablicy liczb całkowitych jest poprawna?

- A. `Type Tablica : array [1..50] of integer;`
- B. `Var Tablica = array [1..50] of integer;`
- C. `Type Tablica = array [1..50] of integer;`
- D. `Var Tablica := array [1..50] of integer;`

**Zadanie 27.**

```
const n = 100;  
type dane = array [0..n] of real;
```

Powyższy zapis:

- A. Deklaruje zmienną typu real
- B. Deklaruje typ tablicowy
- C. Tworzy tablicę wypełnioną liczbami od 1 do 100
- D. Tworzy tablicę wypełnioną liczbami od 0 do 100

**Zadanie 28.**

Za pomocą jakiej procedury zapisujemy dane do pliku:

- A. Read
- B. Write
- C. FileRead
- D. FileWrite

**Zadanie 29.**

Procedura obliczania służy do

- A. znajdowania największej wspólnej wielokrotności.
- B. dzielenia wpisanych liczb z resztą.
- C. obliczenia ciągu arytmetycznego
- D. znajdowania największego wspólnego dzielnika dwóch liczb.

```
procedure procl;  
var  
  n,m,k : integer;  
begin  
  read(m,n);  
  repeat  
    k:=n mod m;  
    n:=m;  
    m:=k  
  until n=0;  
  write(m);  
end;
```

**Zadanie 30.**

Procedurę o nazwie np. kwadrat można w programie:

- A. definiować wielokrotnie
- B. deklarować tylko raz
- C. ukrywać wielokrotnie
- D. wywoływać tylko raz

**Zadanie 31.**

Zmienne użyte w nagłówku procedury lub funkcji w czasie jej deklaracji, to:

- A. zmienne lokalne
- B. parametry formalne
- C. parametry aktualne
- D. zmienne globalne

**Zadanie 32.**

Procedura Rewrite:

- A. tworzy i otwiera plik do odczytu
- B. otwiera plik do zapisu
- C. tworzy i otwiera plik do zapisu
- D. tworzy i otwiera plik do zapisu i odczytu

**Zadanie 33.**

Zmienne deklarowane w procedurze lub funkcji (po słowie VAR), to:

- A. zmienne lokalne
- B. parametry formalne
- C. parametry aktualne
- D. zmienne globalne

**Zadanie 34.**

Co się stanie, jeśli w treści funkcji nie będzie podstawienia wyniku pod nazwę funkcji:

- A. Program z tak zadeklarowaną funkcją się nie skompiluje.
- B. Program się skompiluje, ale nie da się wykonać.
- C. Program da się wykonać, ale funkcja nie będzie zwracała wyniku przez swoją nazwę.
- D. Program się zawiesi.

**Zadanie 35.**

Suma wartości wszystkich komórek tablicy wypełnionej w obok opisany sposób wynosi:

- A. 8
- B. 3
- C. 6
- D. 10

```
var a:array[1..10] of byte;  
begin  
    for i:=3 to 8 do a[i]:=1;  
end.
```

**Zadanie 36.**

Procedura obok jest przykładem

- A. mieszania elementów.
- B. sortowania elementów.
- C. uzupełniania tablicy
- D. wypisywania elementów tablicy

```
procedure sortuj (var T:tablica; n:integer);  
var j, i: integer;  
    pom: real;  
begin  
    for j:=n-1 downto 1 do  
        for i:=0 to j-1 do  
            if T[i]>T[i+1] then  
                begin  
                    pom:=T[i];  
                    T[i]:=T[i+1];  
                    T[i+1]:=pom;  
                end;  
        end;  
    end;
```

**Zadanie 37.**

Chcemy wypełnić tablicę A[1..n] liczbami >10 wczytywanymi z klawiatury, pomijając pozostałe. Która konstrukcja jest niepoprawna?

- A. for i:=1 to n do begin readln(x); if x>10 then A[i]:=x end;
- B. for i:=1 to n do repeat readln(A[i]) until A[i]>10;
- C. i:=1; repeat readln(x); if x>10 then begin A[i]:=x; i:=i+1 end; until i>n;
- D. for i:=1 to n do readln(A[i]);

**Zadanie 38.**

Procedurę o nazwie np. kwadrat można w programie:

- A. wywoływać wielokrotnie
- B. deklarować wielokrotnie
- C. ukrywać wielokrotnie
- D. wywoływać tylko raz

**Zadanie 39.**

Procedura obok jest przykładem

- A. tasowania tali kart.
- B. definiowania zmiennych globalnych.
- C. sortowania kart według kolorów i figur.
- D. rozdawania tali kart między dwóch grających.

```
procedure talie;  
var  
  i,a,b : integer  
  x      : TKarta;  
begin  
  for i:=1 to 1000 do  
  begin  
    a:=1+random(52);  
    b:=1+random(52);  
    x:=talialia[a];  
    talialia[a]:=talialia[b];  
    talialia[b]:=x;  
  end;  
end;
```

**Zadanie 40.**

Co wydrukuje następująca pętla: `for i:=1 to n do writeln(a[i,i])` w przypadku tablicy `a[1..n,1..n]`

- A. główną przekątną tablicy a.
- B. linię ukośną tuż nad główną przekątną tablicy a.
- C. linię ukośną tuż pod główną przekątną tablicy a.
- D. linię ukośną tuż nad główną przekątną tablicy a, ale potem nastąpi błąd wykonania.

**Zadanie 41.**

Zaznacz słowo rozpoczynające deklarację podprogramu.

- A. Program
- B. Begin
- C. Var
- D. Function

**Zadanie 42.**

Zdefiniowana funkcja xxx służy do obliczania:

- A. wartości  $n!$  (silnia)
- B. iloczynu dwóch liczb
- C. wartości wielomianu
- D. wartości pierwiastka z liczby

```
function xxx (n : integer) : integer;  
begin  
  if (n=0) or (n=1) then  
    xxx:=1  
  else  
    xxx:=n*xxx(n-1);  
  end;
```

**Zadanie 43.**

Za pomocą jakiej funkcji zamykamy dostęp do pliku:

- A. CloseFile
- B. Close
- C. Quit
- D. Exit

**Zadanie 44.**

Tablica TAB została zadeklarowana w następujący sposób:

```
const n = 5;  
type tabela = array[-5..n,-5..n] of byte;  
var TAB: tabela;
```

Ile elementów można zapisać w tablicy TAB?

- A. 1
- B. 10
- C. 121
- D. 100

**Zadanie 45.**

```
procedure Instrukcja;  
VAR  
  i: integer;  
BEGIN  
  FOR i:=-5 to 5 DO  
    Writeln('W tym przebiegu pętli wartość i wynosi ',i);  
  END;
```

W procedure przedstawionym powyżej instrukcja Writeln będzie wykonana

- A. 0 razy
- B. 5 razy
- C. 10 razy
- D. 11 razy