

## Test di Verifica AP-04-02-Test

1) Che cosa visualizza il seguente algoritmo?

```
var x,y,z : integer
x := -1
y := 1
z := 2
if !(x>= z and z <= y)
then
  print "ok"
else
  print "ko"
endif
```

- ok
- ko
- l'algoritmo contiene un errore logico
- la prima volta ko, la seconda ok

2) Che cosa visualizza il seguente algoritmo, se si immettessero da tastiera i numeri 5 e 1?

```
var x,y : integer
read x,y
if x >= y+10
then
  print "molto maggiore"
endif
if x <= y-10
then
  print "molto minore"
endif
print "vicino"
```

- "molto maggiore"
- "molto minore"
- "vicino"
- molto minore" "vicino"

3) Che cosa visualizza il seguente algoritmo, se si immettessero da tastiera i numeri 57 e 190?

```
var x,y : integer
```

```
read x,y
if x >= y+10
  then
    print "molto maggiore"
  endif
if x <= y-10
  then
    print "molto minore"
  endif
print "vicino"
```

- "molto maggiore"
- "molto minore"
- "vicino"
- "molto minore" "vicino"

4) Che cosa visualizza il seguente algoritmo, se si immettessero da tastiera i numeri 5 e 1?

```
var x,y : integer
read x,y
if x >= y+10
  then
    print "molto maggiore"
  else if x <= y-10
    then
      print "molto minore"
    else
      print "vicino"
    endif
  endif
endif
```

- "molto maggiore"
- "molto minore"
- "vicino"
- "molto minore" "vicino"

5) Che cosa visualizza il seguente algoritmo, se si immettessero da tastiera i numeri 57 e 190?

```
var x,y : integer
read x,y
if x >= y+10
```

```

then
  print "molto maggiore"
else if x <= y-10
  then
    print "molto minore"
  else
    print "vicino"
  endif
endif

```

- "molto maggiore"
- "molto minore"
- "vicino"
- molto minore" "vicino"

6) Che cosa visualizza il seguente algoritmo, letti da tastiera due numeri interi positivi?

```

var N,M,R : integer
read N,M
R := 0
while N!= 0
  R := R+M
  N := N-1
endwhile
print R

```

- il prodotto di M per N
- il quoziente di M diviso N
- il prodotto di M per (N - 1)
- il quoziente di N diviso M

7) Quando si conosce *a priori* il numero delle iterazioni da svolgere, conviene utilizzare:

- il costrutto repeat until
- il costrutto while
- il costrutto for
- un costrutto ripetitivo qualunque

8) Quale fra i seguenti gruppi di istruzioni producono lo scambio dei valori di due variabili X e Y (Z, X, Y sono variabili dello stesso tipo)?

- X := Y; Z :=X; Y := Z
- Z := Y; X :=Y; X := Z

- $X := Y; Y := X$
- $Z := Y; Y := X; X := Z$

9) Quale delle seguenti affermazioni sul costrutto ripetitivo `repeat until` è vera?

- prima sono eseguite le istruzioni del corpo del ciclo e poi è valutato il predicato di uscita (eseguito il controllo)
- prima viene valutato il predicato di uscita (eseguito il controllo) e poi le istruzioni del corpo del ciclo
- il corpo del ciclo potrebbe non essere mai eseguito
- nessuna delle risposte precedenti è esatta

10) Quale delle seguenti affermazioni è falsa?

- nel `repeat until` le istruzioni del corpo del ciclo sono eseguite se il valore del predicato di uscita è falso
- nel `while` le istruzioni del corpo del ciclo sono eseguite se il valore del predicato di uscita è vero
- nel `for` le istruzioni del corpo del ciclo sono eseguite se il valore della variabile di ciclo è compreso nel suo range (tra il valore iniziale e il valore finale)
- nel `for` le istruzioni del corpo del ciclo sono eseguite almeno una volta nel caso in cui il range della variabile di ciclo è vuoto (per es. `for i=6,5 do..`)

11) Che cosa visualizza il seguente algoritmo, letto da tastiera un numero intero positivo?

```

var K,I,N : integer
K := 1
read N
for I=2,N do
  K := K*I
endfor
print K

```

- il valore della somma  $2 + 3 + \dots + N$
- il fattoriale di  $N$ , cioè il valore  $1 \times 2 \times 3 \times \dots \times N$
- $N - 1$  volte  $K$
- $N - 1$  volte  $K * I$

12) Quante volte viene eseguita l'istruzione `print` nel seguente algoritmo?

```

var J,I,N : integer
read N
for I=1,N do
  for J=1,N do
    print I,J
  endfor
endfor

```

```
        endfor
    endfor
```

- N+N volte
- 2N volte
- N volte
- $N^2$  volte

13) Quante volte viene eseguita l'istruzione print nel seguente algoritmo?

```
var J,I,N : integer
read N
for I=1,N do
    for J=I,N do
        print I,J
    endfor
endfor
```

- N+N volte
- 2N volte
- $N+(N-1)+(N-2)+\dots+2+1$  volte
- $N^2$  volte

# Problemi di programmazione

Risolvere i seguenti problemi descrivendo l'algoritmo in pseudolinguaggio e in Flowchart:

1. Visualizzare i numeri naturali dispari da 3 a 21.
2. Calcolare la somma dei numeri naturali da 10 a 20.
3. Visualizzare il quadrato dei primi 10 numeri naturali.
4. Dato in input un numero intero positivo  $n$ , visualizzare la frase "torno subito" per  $n$  volte. (\*)
5. Trovare il più grande fra tre numeri letti da tastiera.
6. Dato in input il numero  $n$  intero positivo dispari, calcolare la somma  $1 + 3 + 5 + \dots + n$ .
7. Dati in ingresso tre numeri, sviluppare un algoritmo che determini se essi possono essere le lunghezze dei lati di un triangolo rettangolo (consiglio: usare il teorema di Pitagora). (\*)
8. Una società di telecomunicazioni applica le seguenti tariffe telefoniche: a) per i primi 20 minuti di conversazione, si pagano 0,015 € per minuto; b) per i minuti successivi ai primi venti, si pagano 0,011 € per minuto. Sviluppare un algoritmo che, letta da tastiera la durata (in minuti) di una conversazione telefonica, calcoli e visualizzi il costo della telefonata. (\*)

(\*) problemi obbligatori: devono essere necessariamente risolti e inviati al tutor.

## Esercizi a risposta aperta

1) Determinare se le seguenti porzioni di algoritmo sono equivalenti, cioè se risolvono lo stesso problema:

```
R := 1
while R <= 5
  L := 2*pi_greco*R
  A := pi_greco*R^2
  R := R+1
endwhile
```

```
R := 1
repeat
  L := 2*pi_greco*R
  A := pi_greco*R^2
  R := R+1
until R>6
```

2. Spiegare perché la seguente porzione di algoritmo implica un in ciclo infinito. Poi, modificarlo in modo che vengano visualizzati i numeri: 10, 20, .....,190, 200.

```
K := 10
I := 1
repeat
  print I*K
  I := I+1
until K = 20
```

3. Determinare che cosa calcola e visualizza il seguente algoritmo.

```
S := 0
I := 10
A := 3
repeat
  B := A*I
  S := S+B
```

```
    I := I-1
until <= 0
print S
```

4) Completare il seguente algoritmo in modo da visualizzare la sequenza 3, 6, 9, ..., 27, 30.

```
P := .....
N := .....
repeat
    T := P*N
    print T
    P := P+1
until .....
```