

No. Parcial:  
Nombre:  
CI:

# Primer Parcial. Programación 1

Instituto de Computación  
Año 2014

Indique cuál es la salida que produce.

- a) 2 4 8 16 32
- b) 1 4 32
- c) 1 2 4 8 16
- d) 1 4
- e) 2 16 64

Respuesta: <sup>2</sup>

## Observaciones:

- Cada ejercicio tiene una única opción correcta.
- Una respuesta correcta suma 2 puntos
- Una respuesta incorrecta resta 0.5 puntos.
- Un ejercicio sin respuesta no afecta el puntaje.
- Debe entregar la planilla con las respuestas junto con la letra, debe poner nombre completo y cédula en la planilla y en la letra.
- En todos los ejercicios se utiliza el lenguaje Pascal tal como fue dado en el curso.

## Ejercicio 1 Dado el siguiente programa:

```
program ProgMatriz;  
const  
  tope = 3;  
type  
  rango = 1..tope;  
var  
  a : array[rango, rango] of integer;  
  i, j : rango;  
begin  
  for i := 1 to tope do  
  begin  
    a[i,1] := 1;  
    for j := 2 to tope do  
      a[i,j] := a[i,j-1] + 1  
    end;  
  
    for i := 1 to tope do  
    begin  
      for j := 1 to tope do  
        write(a[j,i]);  
        write(' ')  
      end  
    end.  
end.
```

Indique cuál es su salida.

- a) 1-2-3-4-5-6-7-8-9-
- b) 111-222-333-
- c) 1-1-1-2-2-2-3-3-3-
- d) 123-123-123-
- e) 1-2-3-1-2-3-1-2-3-

Respuesta: <sup>1</sup>

## Ejercicio 2 Sea el siguiente programa:

```
program pares;  
const  
  MAX = 20;  
var  
  i,k,result : integer;  
  pp : boolean;  
begin  
  k:= 1;  
  result:= 1;  
  pp:= true;  
  repeat  
    if pp then  
      pp:= not pp  
    else  
      for i:= 1 to k do  
        result:= 2 * result;  
        write(result, ' ');  
        k:= k + 1  
      until result > MAX  
    end.
```

## Ejercicio 3 Suponga la declaración de las siguientes variables:

```
var  
  m, n : integer;  
  x : real;
```

Indique cuál de las siguientes expresiones **no** es de tipo integer.

- a) round(x) \* m
- b) m div n
- c) trunc(sqrt(n)) + m
- d) x \* m
- e) trunc(x/m)

Respuesta: <sup>3</sup>

## Ejercicio 4 Dado el siguiente programa:

```
program ejemplo;  
var n:integer;  
begin  
  readln(n);  
  if n mod 2 = 0 then n:=n+1;  
  if n mod 2 <> 0 then n:=n+3  
  else n:=n+5;  
  writeln(n)  
end.
```

Indique cuál de las siguientes afirmaciones es correcta.

- a) Si se lee el número 10 la salida es 11.
- b) Si se lee el número 20 la salida es 15.
- c) Si se lee el número 35 la salida es 40.
- d) La instrucción n:=n+5 nunca se ejecuta.
- e) La instrucción n:=n+3 nunca se ejecuta.

Respuesta: <sup>4</sup>

**Ejercicio 5** Indique cuál de los siguientes fragmentos de código imprime el máximo entero de una secuencia **no vacía** de enteros positivos ingresados desde la entrada estándar. El fin de la entrada se indica por un entero no positivo.

```
a)  
max:=-1;  
readln(n);  
while n>0 do  
  if n>max then  
    max:=n;  
  writeln(max);
```

```
b)  
max:=-1;  
readln(n);  
while n>0 do  
  if n>max then max:=n;  
  readln(n);  
  writeln(max);
```

```
c)  
readln(max);  
readln(n);  
while n > 0 do begin  
  if n > max then  
    max := n;  
  readln(n)  
end;  
writeln(max);
```

```
d)
max:=-1;
repeat
  readln(n);
  if n>max then max:=n;
until n=max;
writeln(max);
```

```
e)
max:=-1;
fin:=false;
for i:=1 to maxint do begin
  if not fin then begin
    readln(n);
    if n>max then max:=n
    else if n<=0 then fin:=true;
  end
end;
writeln(max);
```

```
program Principal;
var n,x : integer;

procedure primera(n:integer; var y:integer);
begin
  n := n+1;
  y := n
end;

function segunda(x,y:integer) : integer;
begin
  x := x-y;
  segunda := x+2
end;

begin
  n:= 5;
  primera(n,n);
  x := segunda(n,3);
  WriteLn(n, ' ',x)
end.
```

**Respuesta:** <sup>5</sup>

**Ejercicio 6** Dado el siguiente fragmento de código:

```
var
  x, y: integer;

begin
  read(x, y);
  if (x > 0) and (y mod (x-1) = 0)
    write('divisible');

  if (y > 0) and (x mod (y-1) = 0)
    write('divisible')
end.
```

Indique con cuál de las siguientes entradas se imprime divisible dos veces.

- a) 22
- b) 11
- c) 12
- d) 21
- e) 24

**Respuesta:** <sup>6</sup>

**Ejercicio 7** Sea el siguiente programa:

```
program ej;
var
  c : char;
  i : integer;
begin
  i := 0;
  read(c);
  while (c < 'H') or (i <= 3) do begin
    i := i + 1;
    read(c)
  end;
  writeln(c, ' ', i)
end.
```

Indique cuál es la salida que produce al recibir como entrada ATBXCUCO.

- a) X4
- b) U5
- c) T1
- d) U6
- e) T2

**Respuesta:** <sup>7</sup>

**Ejercicio 8** Sea el siguiente programa:

Indique cuál es la salida del programa.

- a) 5 5
- b) 5 6
- c) 6 5
- d) 3 5
- e) 5 3

**Respuesta:** <sup>8</sup>

**Ejercicio 9** Suponga la declaración de las siguientes variables:

```
var
  i : integer;
  b : boolean;
  c : char;
  r : real;
```

Indique cuál de las siguientes asignaciones **no** produce un error de tipos.

- a) i := b
- b) b := c
- c) i := i + r
- d) r := ord(c) / i
- e) c := chr(r)

**Respuesta:** <sup>9</sup>

**Ejercicio 10** Sea la siguiente declaración de variables:

```
const
  M = 7;
var
  a : array [1..M] of integer;
  existe : boolean;
  i, j : integer;
```

Indique cuál de los siguientes fragmentos de código despliega **existe** si y sólo si existe un elemento del array cuyo valor sea igual a la suma de sus dos vecinos (esto es, los elementos anterior y posterior).

```
a)
i := 1;
while (i <= M) and (a[i] <> a[i-1] + a[i+1]) do
  i := i + 1;
if i <= M then writeln('existe');
```

```
b)
for i := 2 to M do
  if a[i] = a[i-1] + a[i+1]
    then existe := true;
if existe then writeln('existe');
```

```

c)
i := 2;
existe := false;
while not existe and (M >= i) do
  if a[i] <> a[i-1] + a[i+1]
  then existe := true
  else i := i + 1;
if existe then writeln('existe');

```

```

d)
i := 2;
j := 1;
while (i < M) and (j <> 0) do
  begin
    j := a[i] - a[i-1] - a[i+1];
    i := i + 1;
  end;
if i < M then writeln('existe');

```

```

e)
i := 2;
while (i < M) and (a[i] <> a[i-1] + a[i+1]) do
  i := i + 1;
if i <> M then writeln('existe');

```

**Respuesta:** 10

**Ejercicio 11** Sea el siguiente fragmento de programa:

```

type
  arreglo = array[0..10, 1..10] of boolean;
var
  a : arreglo;
  c : char;
  i : integer;
  b : boolean;
begin
  c := 'R';
  i := 1;
  b := false;
  ...
end.

```

Indique cuál asignación compila y ejecuta correctamente.

- a) b := b or (2 mod (i-1) = 0)
- b) a[2,0] := (not b) or (i > 0)
- c) a[10, 10] := (chr(i) = c) or (not b)
- d) b := (c = 'R') and (a[0,0] = a[1,1]) and b
- e) a[2,2] := sqrt(i-1) = '1'

**Respuesta:** 11

**Ejercicio 12** Sean las siguientes declaraciones:

```

type
  rango = 0..10;
  indice = 1..9;
  borde = -1..1;
var
  a: rango;
  b: indice;
  c: borde;

```

Indique cuál de las siguientes asignaciones **no** produce un error de ejecución, cualquiera sea el valor de a, b y c.

- a) b:= a;
- b) b:=a+c;
- c) a:=a+c;
- d) c:=c\*(-1);
- e) a:=a+1;

**Respuesta:** 12

**Ejercicio 13** Sea el siguiente programa:

```

Program ej;

type
  enteros = array[1..7] of integer;

var
  i : integer;
  arreglo : enteros;

procedure invertirRango(inicio, fin : integer;
                       var arreglo : enteros);
var medio, aux : integer;
begin
  medio := (inicio+fin) div 2;
  for i := inicio to medio do begin
    aux := arreglo[i];
    arreglo[i] := arreglo[(medio-i)+medio];
    arreglo[(medio-i)+medio] := aux;
  end;
end;

begin
  for i := 1 to 7 do
    read(arreglo[i]);

  invertirRango(4, 7, arreglo);
  for i := 1 to 7 do
    write(arreglo[i]);

  invertirRango(4, 12, arreglo);
  for i := 1 to 7 do
    write(arreglo[i])
end.

```

Indique cuál es la salida del programa si la entrada es:

2 3 4 5 2 5 6

- a) 2345256 y luego da error
- b) 2346525 y luego da error
- c) 2346525 y 2345256
- d) Se produce un error en la primer invocación al procedimiento
- e) 2346525 y 6525432

**Respuesta:** 13

**Ejercicio 14** Dado el siguiente programa:

```

program ProFun(output);

var x, y, z : integer;

procedure calcula(y1 : integer; var y2, y3 : integer);
var y4 : integer;
function valor(x1, x2 : integer) : integer;
var x3 : integer;
begin
  x3 := x2;
  x1 := x3;
  valor := x1 + x2 + x3;
end;
begin
  y4 := y2;
  y1 := y3 + y4;
  y2 := y1 + y4;
  y4 := 9;
  y3 := valor(y2, y4);
end;

begin
  x := 1;
  y := 2;
  z := 3;
  calcula(x, y, z);
  writeln('=', x, '=', y, '=', z);
end.

```

Indique cuál es la salida.

- a) =5=7=27
- b) =1=2=3
- c) =1=7=3
- d) =1=2=27
- e) =1=7=27

**Respuesta:** 14

**Ejercicio 15** Dada la siguiente declaración de variables

```
var
  i1, i2, i3 : Integer;
  r1 : real;
  c1 : char;
  prueba:boolean;
```

y la siguiente declaración de función:

```
function parcial(uno, dos : integer; tres,
  cuatro : real; cinco, seis : char) : boolean;
```

Indique cuál de las siguientes invocaciones es correcta.

- a) prueba := parcial(i1, i2, i3, r1, c1, chr(i1))
- b) prueba := parcial(i1, r1, i3, r1, c1, chr(i1))
- c) prueba := parcial(i1, i2, i3, r1, c1, ord(c1))
- d) prueba := parcial(i1, r1, i3, r1, c1, ord(i1))
- e) prueba := parcial(i1, i2, i3, r1, c1, chr(r1))

**Respuesta:** 15

**Ejercicio 16** Dada la siguiente declaración de variables:

```
var x : real; y : integer;
```

Indique cuál de los siguientes fragmentos de código compila y ejecuta sin errores para cualquier valor de x e y.

```
a)
if (x / y = 1) then
  writeln('Son iguales')
```

```
b)
if (y <> 0) then begin
  y := x * y;
  if (y = 1) then
    writeln('Son inversos')
end
```

```
c)
if (x / y = 1) and (y <> 0) then
  writeln('Son iguales')
```

```
d)
if not ((y = 0) or (x * y <> 1)) then
  writeln('Son inversos')
```

```
e)
if (y <> 0) or (x / y = 1) then
  writeln('Son iguales')
```

**Respuesta:** 16

**Ejercicio 17** Sea el siguiente programa.

```
program prog;
var a, b : integer;

procedure proc(m: integer; var n: integer);
var a: Integer;
begin
  m := m+2;
  a := m*n;
  n := a+1
end;

begin
  a := 0;
  b := 5;
  proc(a, b);
  writeln(a, ' ', b)
end.
```

Indique cuál es la salida del programa.

- a) 2 11
- b) 10 11
- c) 0 5
- d) 10 5
- e) 0 11

**Respuesta:** 17

**Ejercicio 18** Sea el siguiente programa.

```
program prog;
var y : integer;

function f(x : integer; y : integer):integer;
  function g(z : integer):integer;
  var a : integer;
  begin
    { sentencias de g }
  end;
begin
  { sentencias de f }
end;

procedure p(var x: integer);
begin
  { sentencias de p }
end;

begin
  { sentencias del programa principal }
end.
```

Indique cuál de las siguientes afirmaciones es correcta.

- a) Se puede invocar a g en las sentencias del programa principal.
- b) En las sentencias de g se puede modificar la variable global y.
- c) Se puede invocar p en las sentencias de f.
- d) Se puede invocar a f en las sentencias de p.
- e) En las sentencias de p se puede modificar la variable local a de g.

**Respuesta:** 18

**Ejercicio 19** Sea la siguiente declaración de variables:

```
var x, y, z : integer;
```

Indique cuál de los siguientes fragmentos de código deja en x el mínimo valor de x,y,z y en z el máximo valor de x,y,z. El valor con que finalice la variable y es irrelevante.

```
a)
if (x <= y) and (z <= y) then z := y;
```

```
b)
if (x > y) and (x-y = 0)
then begin
  y := z;
  z := x
end;
```

```
c)
if x = y+z
  then if x-y = z
        then begin
              y := x;
              x := z;
              z := x;
            end;
```

```
d)
res := -1;
i := 1;
while ((arr[i] <> x) and (i < N)) do
  i := i + 1;
if (arr[i] = x) then
  res := i;
writeln(res);
```

```
d)
if x <= y
  then if (y > z) and (x <= z)
        then z := y
        else x := z;
```

```
e)
res := -1;
i := 0;
repeat
  i := i + 1;
until ((i <= N) and (arr[i] <> x));
if (arr[i] = x) then
  res := i;
writeln(res);
```

```
e)
if y < x then
  begin
    x:= x + y;
    y:= x - y;
    x:= x - y;
  end;
if z < x then
  begin
    x := z;
    z := y;
  end
else if z < y then
  z := y;
```

Respuesta: <sup>20</sup>

## Respuestas

- 1b)
- 2b)
- 3d)
- 4d)
- 5c)
- 6a)
- 7b)
- 8c)
- 9d)
- 10e)
- 11c)
- 12d)
- 13a)
- 14e)
- 15a)
- 16d)
- 17e)
- 18d)
- 19e)
- 20d)

Respuesta: <sup>19</sup>

**Ejercicio 20** Dadas las siguientes declaraciones:

```
const
  N = 10;

type
  rango = 1 .. N;

var
  arr : array [rango] of char;
  i : rango;
  res : integer;
  x : char;
```

Indique cuál de los siguientes fragmentos de código imprime el **menor** índice del arreglo que contiene el valor de la variable x. Si no lo contiene se imprime -1. Asuma que las variables x y arr ya fueron inicializadas.

```
a)
res := -1;
for i := 1 to N do
  if (arr[i] = x) then
    res := i;
writeln(res);
```

```
b)
res := -1;
i := 1;
while ((i < N) and (arr[i] <> x)) do
  i := i + 1;
res := i;
writeln(res);
```

```
c)
res := -1;
i := 1;
while ((i <= N) and (arr[i] <> x)) do
  i := i + 1;
res := i;
writeln(res);
```