

PROGRAMACION 1 TT

Guia de estudios 1:

0)

1. Establecer cual es la diferencia entre un compilador y un intérprete.
2. Diferenciar lenguajes de alto nivel y lenguajes de bajo nivel.
3. Explicar qué es un algoritmo y qué características debe cumplir.
4. Dar un ejemplo de algoritmo ¿cómo podría probarse que funciona?
5. Para cada uno de los siguientes tipos de datos, caracterizar cada uno e indicar cuales son los operadores permitidos qué hace cada uno:

int
float
char

6. Indicar qué tipos de int y float hay en C.
7. Mencione dos funciones distintas provistas por el estándar de C para la entrada de datos, y dos funciones para la salida.

1) Indicar qué es lo que se almacena en cada una de las variables indicadas a continuación, si se realizan las operaciones que se indican, sabiendo que se han definido las variables usadas de la siguiente forma, e indicándose sus valores iniciales:

int a=3,b=2,c=1, d,e;

float m=2.5, n=5.4, r,s;

- a) d= m;
- b) d=n;
- c) e=3.7;
- d) d=a;
- e) d=a+b;
- f) d=a*b;
- g) d=a/b;
- h) d=b/4;
- i) d=a%b;
- j) d=a%2;
- k) e=b/c;
- l) e=c/b;
- m) r=a+b;
- n) r=a/b;
- o) r=a/2;
- p) r=a/2.0;
- q) s=m+n;
- r) s=3+4-1;
- s) s=3.0+4.0-1;
- t) a=m;
- u) a=m/2;
- v) a=m/2.0;
- w) a=3.0+4.0-1;
- x) r=a+1;
- y) r=a+1.0;
- z) a++;
- aa) r++;
- bb) b - -;
- cc) a+=5;
- dd) s*=5;

2) Considere el siguiente programa:

```
#include <stdio.h>
int main()
{
    printf( " Me gusta C" );
    printf (" Fin del mensaje \n");
    printf ("Adios");
    return(0);
}
```

a) Indique cual es el efecto de incorporar a la segunda salida por pantalla el carácter '\n'. Verifique el uso incorporándolo a otras sentencias de emisión por pantalla.

b) Investigue, en las salidas, el uso de: '\t',

c) ¿cuál es el sentido de la sentencia return 0 de la última línea?

3) Indicar qué valor se almacena en cada variable de la izquierda de cada asignación, siendo:

```
int  a, b, c, d =0, e=1;
//recordar que en c, 0  corresponde a falso.
```

```
int a=2, b=3, c;
a)  a= 5>3;
b)  b=( (4<5) && (2>2));
c)  a=!b;
d)  c= ((a%2 ==0) || (a-b <10));
e)  c= (d && e);
f)  c=(d || e);
g)  c= !(a && b);
h)  c= (! a) && (!b);
i)  c= (a && (!a));
j)  c=(((a<=(b*3.2)) &&(b%2!=0))|| (1));
```

4) Indicar cuál es la salida por pantalla para cada sentencia printf, siendo:

```
int a=15;
float b=13.546;
char c='A';
char d='a';
int e=5>1;
a)  printf ("a vale %d", a);
b)  printf ("a vale %o", a);
c)  printf ("a vale %X", a);
d)  printf ("a vale %f", a);
e)  printf ("a vale %c", a);
f)  printf ("b vale %d", b);
g)  printf ("b vale %5.2d", b);
h)  printf ("b vale %f", b);
i)  printf ("b vale %.1f", b);
j)  printf ("b vale %.2f", b);
k)  printf ("b vale %6.4f", b);
l)  printf ("b vale %6.1f", b);
m)  printf ("b vale %c", b);
n)  printf ("b vale %o", b);
o)  printf ("b vale %X", b);
p)  printf ("c vale %d", c);
```

- q) printf ("c vale %f", c);
- r) printf ("c vale %c", c);
- s) printf ("c vale %o", c);
- t) printf ("c vale %X", c);
- u) printf ("d vale %d", d);
- v) printf ("d vale %f", d);
- w) printf ("d vale %c", d);
- x) printf ("d vale %o", d);
- y) printf ("d vale %X", d);
- z) printf ("e vale %d", e);
- aa) printf ("e vale %f", e);
- bb) printf ("e vale %c", e);
- cc) printf ("e vale %o", e);
- dd) printf ("e vale %X", e);

5) Indicar qué queda almacenado en la variable con la que se ingresan datos en cada caso, si se han definido las variables de la siguiente forma:

```
int a;
float b;
char c;
```

- a) scanf ("%d", &a);
- b) scanf ("%f", &a);
- c) scanf ("%c", &a);
- d) scanf ("%d", &b);
- e) scanf ("%f", &b);
- f) scanf ("%c", &b);
- g) scanf ("%d", &c);
- h) scanf ("%f", &c);
- i) scanf ("%f", &c);
- j) A partir de las situaciones anteriores, sintetice el uso de estos especificadores para la entrada %d, %o, %b, %x, %c, %f

6) Indique funciones del preprocesador, compilador y enlazador. Especifique salida de cada uno.

7) Explique la finalidad de las líneas #include<....h> de la parte superior del archivo. ¿Qué ocurre si se omite el header adecuado?

8) Considere este esquema de programa y

- a) Coloque el/los headers adecuados
- b) Complete el programa para que se emita el perímetro y la superficie del cuadrado cuyo lado se ha ingresado.
- c) Investigue qué sucede si se coloca un especificador equivocado en el ingreso de datos.
- d) Observe qué ocurre si se omite el & en la sentencia de ingreso de datos

```
int main()
{
float lado, perimetro, superficie;
printf ("Ingrese la medida del lado de un cuadrado:");
scanf ("%f", &lado);
// completar aquí
return 0;
}
```