



PRACTICA DEL MODULO 1

Ejercicios

Breve descripción

Ejercicios de programación para el módulo 1.

Versión 1.1

Martin Jerman

Martin.jerman@inspt.utn.edu.ar



Recomendaciones

Comprende a fondo el problema antes de delinear la solución

1. Establece cuales son los datos, si los conoces o no, y si puedes averiguarlos de algún modo.
2. Analiza las condiciones que deban ser tenidas en cuenta.
3. Planea la solución.
4. Para orientarte, considera las siguientes preguntas:
 - a. ¿Es conocido el problema?
 - b. Si es desconocido: ¿conoce la solución de alguno similar?
 - c. ¿Puede resolver el problema en su totalidad?
 - d. Si no es así: ¿Puede resolver parte del problema?
 - e. Si no es así: ¿Puede resolverlo en distintas condiciones?
5. Es importante que desarrolles todo lo que puedas de la solución y que determines con claridad cuál es la dificultad que no puedes salvar. Explicitarla, en muchos casos, orienta hacia la solución de esta.
6. Desarrolla el plan de la solución (algoritmo), chequeando cada paso.
7. Examina la solución en su conjunto. Recién entonces, escribe el código.

Variables, operadores y tipo de datos. Especificadores de formato. Entrada y salida estándar

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 en forma consecutiva:

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

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

d=m;	e=b/c;	s=3.0+4.	r=a+1.0;
d=n;	e=c/b;	0-1;	a++;
e=3.7;	r=a+b;	a=m;	r++;
d=a;	r=a/b;	a=m/2;	b--;
d=a+b;	r=a/2;	a=m/2.0;	a+=5;
d=a/b;	r=a/2.0;	a=3.0+4.	s*=5;
d=a%b;	s=m+n;	0-1;	
d=a%2;	s=3+4-1;	r=a+1;	

2. Indicar qué valor se almacena en cada variable de la izquierda de cada asignación, siendo:

```
int a, b, c, d =0, e=1; //0 corresponde a falso
```

```
int x=2, y=3;
```

1) a= 5>3;	7) c=!(d && e);
2) b=((4<5) && (2>2));	8) c=(! d) && (!e);
3) a=!e;	9) c=(a && (!a));
4) b=((x%2 ==0) (x-y <10));	10) c=(((x<=(y*3.2))
5) c=(d && e);	&&(y%2!=0)) (1));
6) c=(d e);	

3. Indicar cuál es la salida por pantalla para cada sentencia printf, siendo:

```
int a=5, e=5>1;
```

```
float b=13.546;
```

```
char c='A';
```

```
char d='a';
```

1) printf ("a vale %d", a);	3) printf ("a vale %X", a);
2) printf ("a vale %O", a);	4) printf ("a vale %f", a);



```

5) printf ("a vale %c", a);
6) printf ("b vale %d", b);
7) printf ("b vale %5.2d", b);
8) printf ("b vale %f", b);
9) printf ("b vale %.1f", b);
10) printf ("b vale %.2f", b);
11) printf ("b vale %6.4f", b);
12) printf ("b vale %6.1f", b);
13) printf ("b vale %c", b);
14) printf ("b vale %O", b);
15) printf ("b vale %X", b);
16) printf ("c vale %d", c);
17) printf ("c vale %f", c);
18) printf ("c vale %c", c);
19) printf ("c vale %O", c);
20) printf ("c vale %X", c);
21) printf ("d vale %d", d);
22) printf ("d vale %f", d);
23) printf ("d vale %c", d);
24) printf ("d vale %O", d);
25) printf ("d vale %X", d);
26) printf ("e vale %d", e);
27) printf ("e vale %f", e);
28) printf ("e vale %c", e);
29) printf ("e vale %O", e);
30) printf ("e vale %X", e);
    
```

4. 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;
    
```

```

1) scanf ("%d", &a);
2) scanf ("%f", &a);
3) scanf ("%c", &a);
4) scanf ("%d", &b);
5) scanf ("%f", &b);
6) scanf ("%c", &b);
7) scanf ("%d", &c);
8) scanf ("%f", &c);
9) scanf ("%f", &c);
    
```

5. ¿Cuáles de los siguientes son tipos válidos?

```

1) unsigned char
2) long char
3) unsigned float
4) double char
5) signed long
6) unsigned short
7) signed long int
8) long double
9) long bool
    
```

6. Supongamos que tenemos estas variables:

```

int a=10;
float b=19.3;
double d=64.8;
char c=64;
    
```

Indica el tipo de dato del resultado para las expresiones siguientes:

a) a+b	b) c+d	c) (int)d+a	d) d+b	e) (float)c+d
char	char	char	char	char
int	int	int	int	int
float	float	float	float	float
double	double	double	double	double

Operador sizeof()

7. Dado el siguiente programa:

```

int main() {
    char c;
    short s;
    int i;
    long l;
    float f;
    double d;
    long double ld;
    }
    
```



```

printf("\nA continuación...\n");
printf("El sizeof de un char es %d\n", sizeof(c));
printf("El sizeof de un short es %d\n", sizeof(s));
printf("El sizeof de un int es %d\n", sizeof(i));
printf("El sizeof de un long es %d\n", sizeof(l));
printf("El sizeof de un float es %d\n", sizeof(f));
printf("El sizeof de un double es %d\n", sizeof(d));
printf("El sizeof de un long double es %d\n", sizeof(ld));
system("Pause");
return 0;
}

```

Interprete la salida del programa y responda: qué es lo que hace el operador `sizeof()`?

Operadores de asignación compacta

8. Suponiendo los siguientes valores iniciales para las variables (int):

```

x = 2;           z = 9;           s = 10;
y = 6;           r = 100;        a = 15; b = 3;

```

¿Cuáles son los valores correctos en cada expresión?

a) <code>x += 10;</code> [] 12 [] 10 [] 11	c) <code>r /= 0;</code> [] infinito [] 1 [] error
b) <code>s *= b;</code> [] 9 [] 13 [] 30	d) <code>y += x + 10;</code> [] 8 [] 12 [] 18
	e) <code>z -= a*b;</code> [] -36 [] -18 [] 36

9. Usa expresiones equivalentes para las siguientes, usando operadores mixtos.

a) <code>x = 10 + x - y;</code> <code>x += 10-y</code> <code>x -= y+10</code> <code>x += 10+y</code>	c) <code>y = y/(10+x);</code> <code>y /= 10*x</code> <code>y /= 10 + y/x</code> <code>y /= 10+x</code>
b) <code>r = 100*r;</code> <code>r *= 100*r</code> <code>r *= 100</code> <code>r += 100</code>	d) <code>z = 3 * x + 6;</code> <code>z += 6</code> <code>z *= 3</code> no es posible

10. Coloca los signos de puntuación donde correspondan.

```

#include <stdio.h>
main(){
    int pies pulgadas
    pies = 5
    pulgadas = pies * 12
    printf("%d" pulgadas)
}

```

11. Siendo "y" una variable int y "x" una variable float, indica si son correctas las sentencias:

```

x = .25 + y
y = 0.25 + x
y = 0,25 + x

```

12. Evalúa cada una de las siguientes expresiones. Luego comprueba el resultado en el laboratorio.

1) <code>8 * 6 / 3 * 4</code>	5) <code>(8 * 6) / (3 * 4)</code>
2) <code>(8 * 6) / 3 * 4</code>	6) <code>1 + 4 * 5 + 8 / 4 + 4</code>
3) <code>8 * 6 / (3 * 4)</code>	7) <code>1 + 4 * 5 - 8 / 4 + 4</code>
4) <code>(8 * 6 / 3 * 4)</code>	



13. Supone que las variables a, b y c tienen asignados los valores 49, 5 y 3 respectivamente.

Encuentra:

1) $a \% b * c + 1$

2) $a \% (b * c) + 1$

3) $24 / c * 4$

4) $(int) (a / b) \% 2$

5) $48 / (c * 2) * 4$

14. Describe los tipos de errores que se pueden encontrar en un programa y en qué momento son descubiertos.

15. Desarrolla un algoritmo que permita leer 2 valores y emitir por pantalla la suma de los dos, la resta, producto, división, promedio y el doble producto del primero menos la mitad del segundo.

16. Encuentra el error en cada uno de los siguientes programas e indica de qué tipo es.

a)

```
main(){
    integer x;
    real y;
    scanf("%D", y);
    printf("%f",x);
}
```

b)

```
main(){
    int n,total;
    float promedio;
    n=0;
    promedio=total/n;
    printf("prom=%f\n",promedio);
}
```

17. Desarrolla un algoritmo que permita, dados ciertos centímetros como entrada de tipo flotante, emitir por pantalla su equivalencia en pies (enteros) y en pulgadas (flotante, 1 decimal).
18. Construye un programa que pregunte los años que tienes y emita como respuesta el número de días vividos.
19. Construye un programa que dados el valor de 1 kg de determinada mercadería y la cantidad mercadería comprada, emite el valor del total a pagar.
20. Construye un programa que permita ingresar los valores de 2 de los ángulos interiores de un triángulo, y se emita por pantalla el valor del restante.
21. Construye un programa que permita ingresar las medidas de los lados de un rectángulo; el mismo debe emitir por pantalla su superficie y su perímetro.
22. Construye un programa que permita ingresar la superficie de un cuadrado (en m2), el mismo debe emitir por pantalla su perímetro.

Uso de la librería math.h

23. Verifica el siguiente programa y extrae conclusiones:

```
#include <stdio.h>
#include <math.h>
```

```
int main(){
    /* calcula y despliega la raiz cuadrada */
    printf("sqrt(%.1f) = %.1f\n", 9.0, sqrt( 9.0 ));
}
```



```
/* calcula y despliega la función exponencial e a la x */
printf( "exp(%.1f) = %f\n", 2.0, exp( 2.0 ) );

/* calcula y despliega el logaritmo the logorithm (base e) */
printf( "log(%.1f) = %.1f\n", 7.389056, log( 7.389056 ) );

/* calcula y despliega el logaritmo (base 10) */
printf( "log10(%.1f) = %.1f\n", 100.0, log10( 100.0 ) );

/* calcula y despliega el valor absoluto */
printf( "fabs(%.1f) = %.1f\n", 13.5, fabs( 13.5 ) );

/* calcula y despliega el entero mas cercano hacia arriba */
printf( "ceil(%.1f) = %.1f\n", 9.2, ceil( 9.2 ) );

/* calcula y despliega el entero mas cercano hacia abajo */
printf( "floor(%.1f) = %.1f\n", -9.8, floor( -9.8 ) );

/* calcula y despliega la potencia */
printf( "pow(%.1f, %.1f) = %.1f\n", 2.0, 7.0, pow( 2.0, 7.0 ) );

/* calcula y despliega sin( x ) */
printf( "sin(%.1f) = %.1f\n", 0.0, sin( 0.0 ) );

/* calcula y despliega cos( x ) */
printf( "cos(%.1f) = %.1f\n", 0.0, cos( 0.0 ) );

/* calcula y despliega tan( x ) */
printf( "tan(%.1f) = %.1f\n", 0.0, tan( 0.0 ) );

system( "pause" );
return 0; /* indica terminación exitosa */
} /* fin de main */
```

24. Construye un programa que permita ingresar 2 tiempos, expresados en horas, minutos y segundos, el mismo debe emitir por pantalla la suma de ambos (también en horas, minutos y segundos).

Directiva #define

25. Transcribe el siguiente programa y extrae conclusiones:

```
#include <stdio.h>
#include <stdlib.h>
#define CUAD(x) (x*x) /* Definición de macros */
int main()
{
    float a;
    printf("\nEscriba un numero: ");
    scanf("%f",&a);
    printf("\nSu cuadrado es: %.2f\n",CUAD(a));
    system("pause");
    return 0;
}
```

Ejercicios integradores

26. Desarrollar un algoritmo que le permita leer un valor para el radio (R) y calcule el área (A) de un círculo y emitir su valor.



27. Determinar la hipotenusa de un triángulo rectángulo conocidas las longitudes de sus dos catetos. Desarrollar los correspondientes algoritmos.
28. Dada una cantidad entera de segundos (menor a 86400) y convertirla en horas, minutos y segundo utilizando los operadores de cociente y resto enteros.
29. Dada una fecha en el formato DDMMAAAA y separarlo en Día, Mes y Año utilizando operaciones aritméticas.
30. Desarrollar un algoritmo que permita leer un valor que represente una temperatura expresada en grados Celsius y convierta dicho valor en un valor expresado en grados Fahrenheit.
31. Desarrollar un algoritmo que permita calcular el área de un triángulo en función de las longitudes de sus lados previamente leídos desde el teclado.
32. Desarrollar un algoritmo que permita determinar el área y volumen de un cilindro cuyo radio (r) y altura (h) se leen desde teclado.
33. Desarrollar un algoritmo que permita calcular el área (a) de un segmento de círculo. Análisis: Para calcular el área de un segmento de círculo lo primero que hay que hacer es leer el valor del radio y leer el valor de x que es la distancia del centro al segmento. Una vez leído dichos valores se calcula aplicando la fórmula respectiva y por último se emite el valor del área.