TIPOS DE DATOS BASICOS EN LENGUAJE C

TIPO	ANCHO EN BIT	RANGO EN PC
char	8	-128 a 127
int	16	-32768 a 32767
float	32	3.4E-38 a 3.4E+38
double	64	1.7E-308 a 1.7E+308
void	0	sin valores

TIPOS DE DATOS

TIPO	ANCHO EN BIT	RANGO EN PC
char	8	-128 a 127
unsigned char	8	0 a 255
signed char	8	-128 a 127
int	16	-32768 a 32767
unsigned int	16	0 a 65535
signed int	16	-32768 a 32767
short int	16	-32768 a 32767
unsigned short int	16	0 a 65535
signed short int	16	-32768 a 32767
long int	32	-2147483648 a 2147483647
signed long int	32	-2147483648 a 2147483647
unsigned long int	32	0 a 4294967295
float	32	3.4E-38 a 3.4E+38
double	64	1.7E-308 a 1.7E+308
long double	64	1.7E-308 a 1.7E+308

COMBINACIONES DE TIPOS DE DATOS

El tamaño en bits asignado al tipo de dato que se use depende de la capacidad de la computadora y del compilador utilizado. Ejemplo: en una PC normal con un compilador de TC ver 2.0, un tipo *int* es de 16 bits, para una AIX-RISC system6000 un tipo *int* es de 32 bits.

DECLARACIONES EN C.

Una declaración asocia un tipo de datos determinado a un grupo de variables.

ejemplos:

```
int a,b,c;
char d,e;
float f,g;
long int h,i,j;
double k,l,m;
```

CONSTANTES EN C.

Ejemplos de constantes

```
#define pi 3.1415
#define verdad 1
const int a=3;
```

MACROS EN C

La forma de una macro es ***#define nombre texto de reemplazo***. El nombre es la macro a sustituir, el texto de reemplazo es una línea normalmente, pero puede continuarse en varias líneas utilizando un al final de cada línea para indicar que se continua.

ejemplo sin argumentos:
#define infinito for(;;)
define una nueva palabra, infinito, para un ciclo infinito.

ejemplo con argumentos:
#define multiplica(a,b) (a*b)

la línea x=multiplica((3+4),(4+6));

será reemplazada por la línea x=((3+4)*(4+6));

OPERADORES

Operadores aritméticos: + - * % (modulo)

Operadores unitarios:

negación : !
menos : -2 -(a+b) -0.34

incremento: ++j j++

decremento: --j j—

ejemplo:

para j=1	SALIDA	para j=1	SALIDA
printf("j=%dn",j);	1	printf("j=%dn",j);	1
printf("j=%dn",++j);	2	printf("j=%dn",j++);	1
printf("j=%dn",j);	2	printf("j=%dn",j);	2

Operador sizeof(tamaño de):

Devuelve el tamaño de su operando en bytes, ejemplo:

printf("Entero tiene : %d\n bytes",sizeof(int));

Operadores relacionales:

<	(menor que)
<=	(menor o igual que)
>	(mayor que)
>=	(mayor o igual que) ejemplo
==	(idéntico)
!=	(diferente)

Operadores lógicos:

&& (AND) 11 (OR)

Operadores de asignación:

Los operadores como +=, -=, =, *=, %=, son operadores de asignación.

Expresiones tales como	Se escriben de la forma
i = i + 2	i+=2
j = j - 3	j-=3
k = k /5	k/=5
n = n * 6	n*=6
m = m % 8	m%=8

Operador condicional:

Sustituye la expresión if-else, y se simplifica el código.

expresión 1 ? proposición 2: proposición 3

ejemplo: f=(a>b) ? a-b : a+b;

en donde a-b se ejecuta si a>b es verdad, de lo contrario se ejecutara a+b

ESTRUCTURA GENERAL DE UN PROGRAMA EN C

A continuación se muestra un bosquejo de como puede ser la estructura de un programa en C.

```
/*comentarios sobre lo que el programa realiza */

/*declaración de archivos de encabezado*/
#include <archivo.h>

/*constantes*/
nombre_constantes;

/*plantillas de estructuras */
struct nombre{declaradores}

/*declaración de funciones */
tipo nombre_funcion1(parametros);
...

/*variables globales*/
tipo nombre_variable;

/*función principal */

void main(void)
{
/*Declaración de variables locales */
tipo nombre_variable;

/*contenido interno */

nombre_funcion1(argumentos);
...
...
...
}

/*cuerpo de las funciones */
tipo nombre_funcion1(argumentos)
{
/*variables locales a la función */
tipo nombre_variable;
/*contenido interno*/
...
}
```

FUNCIONES DE BIBLIOTECA ANSI C

Kernighan y Ritchie en su apéndice B muestra la biblioteca definida por el estándar ANSI C y el contenido de las funciones dentro de estas, se puede tener acceso a un header por medio de #include <header>. A continuación se da un resumen de lo que hace cada archivo de cabecera.

<assert.h> Define la macro assert(), es muy adecuada en depuración. Se le da un argumento que es una expresión que se afirma que es verdadera. El preprocesador genera código que comprobara la afirmación. Si es falsa, el programa se detendrá después de dar un mensaje de error en el que se indique cual era la afirmación y que esta era errónea.
<ctype.h> Definición de varios tipos de macros, permite el manejo de tipos en el lenguaje.
<errno.h> Maneja errores de código para registrar razones de error.
<float.h> Define implementaciones especificas de macros para el trato con números flotantes.
<limits.h> Define implementaciones especificas de limites sobre tipos de valores.
<locale.h> Declara las funciones, tipos y macros relacionados al formato de valores numéricos.
<math.h> Declara funciones y macros matemáticas.
<setjmp.h> Proporcionan una forma de evitar la secuencia normal de llamadas y regreso de funciones, típicamente para permitir un regreso inmediato de una llamada a una función profundamente anidada.
<signal.h> Da facilidades para manejar condiciones excepcionales que aparecen durante la ejecución, tal como una señal de interrupción de una fuente externa o un error en la ejecución.
<stdarg.h> Proporciona recursos para recorrer una lista de argumentos de función de tamaño y tipo desconocido.
<stddef.h> Definición de tipos comunes NULL, errno, ptrdiff_f, size_t.
<stdio.h> ENTRADA Y SALIDA DE FLUJOS. Las funciones, tipos y macros de entrada y salida con o sin formato están aquí, todo lo relacionado a archivos.
<stdlib.h> Definiciones para tipos comunes, variables y funciones que se relacionan con dirección de memoria, ordenamiento, búsqueda, conversión de cadenas, aritmética entera.
<string.h> Definiciones de memoria y funciones de cadena de caracteres.
<time.h> Declaraciones de funciones y estructuras que relacionan al tiempo.

TAREA: Hacer una tabla como la presentada al inicio para los tipos de datos y las combinaciones de tipos de datos. Que incluya en la primera columna el tipo/combinación, y el ancho en bit. Aplicado al servidor de la escuela de la Licenciatura en Ciencias Genómicas.

TAREA OPCIONAL: Hacer una tabla como la presentada al inicio para los tipos de datos y las combinaciones de tipos de datos. Que incluya en la primera columna el tipo/combinación, el ancho en bit y el rango. Aplicado al servidor de la escuela de la Licenciatura en Ciencias Genómicas.