**Repaso de programacion**

Lectura: scanf(“%formato”,&variable); letra=getchar();

Escritura: printf(“%formato”,variable; ejemplo printf( “%d”,numero); putchar(letra);

**Declaracion de variable**

Tipo nombre = valor; ejemplo int numero=5;

**Declaracion de función**

Tipo nombre\_func(lista de parámetros

{

Declaraciones;

Proposiciones;

}

**Declaracion de arreglos**

Tipo nombre [tamaño]; ejemplo int arreglo[50]={1,2,3,4,5,6,7,8,9,……,50};

Para mostrar un elemento en particular: printf(“arreglo[espacio2]= %d”,arreglo[1]) == mostrara “2”. notar que el conteo comienza desde 0-49

Para mostrar el arreglo entero se necesita una función que escriba cada numero individual.

**Declaracion de puntero**

Tipo \*nombre-variable-puntero;

Tipo: tipo de c(int,char,float)

\*:operador de puntero. (decara como puntero, no como contenido)

nombre-variable-puntero: indentificador.

\*:operador de indireccion; &operador de dirección.

Ejemplo: int numero=5;

P1=&numero; [recibe la dirección de la variable “numero”]. Al imprimir P1 no mostrara 5, sino que mostrara la DIRECCION de memoria de la variable “numero”.

Ejemplo de clase.

int main()

{

int prim=10, seg=20,temp;

int \*punt\_num; //crea una variable punt\_num que TAMBIEN es un puntero

printf("Al inicio - prim:%d, seg:%d",prim,seg);

punt\_num = &prim; //llena a punt\_num con la direccion de memoria de prim (llena con AA)

temp = \*punt\_num; //llena a temp con lo que sea que este contenido en la direccion de punt\_num(llena con 10)

\*punt\_num = seg; //llena el espacio de memoria de punt\_num(AA) con el valor de seg (20) (AA pasa a contener 20)

seg = temp; //llena la variable seg (20) con el valor de temp(10) (seg ahora vale 10)

printf("Despues - prim:%d, seg:%d",prim,seg);

return 0;

}