



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Programas de cómputo

Materia de Programación I

Profesor

M.C. Yolanda Moyao Martínez

Fecha

Enero 2017

1. Programa que muestra un arreglo de forma descendente

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

int main(void)
{
    int x, tabla[100];

    for (x=1; x<=100; x++)
    {
        tabla[x]=x;
    }

    for (x=1; x<=100; x++)
    {
        printf("%d\n", tabla[x]);
    }

    system("PAUSE");
    return 0;
}
```

2. Programa que compara dos números:

```
#include <stdio.h>
int main ()
{
    int a,b;
    printf ("\n Dame un valor: ");
    scanf ("%d",&a);
    printf ("\n Dame un valor: ");
    scanf ("%d",&b);
    /*
if (a>b)
    printf ("\n el primer numero es mayor");
else
{
    if (a<b)
        printf ("\n El segundo numero es mayor");
    else
        printf ("\n Son iguales");
    }
*/
if (a==b)
    printf ("\n Son iguales");
else
    a>b ? printf ("\n el primer numero es mayor") : printf ("\n El segundo
numero es mayor");
return 0;
}
```

3. Programa que simula una calculadora con operaciones básicas:

```
#include <stdio.h>
int main ()
{
    int a,b,c;
    float d;
    char opc;
    printf ("\nDame el primer dato:");
    scanf ("%d",&a);
    printf ("\nDame el segundo dato:");
    scanf ("%i",&b);
    getchar();
    printf ("1.Suma\n2.Resta\n3.Multiplicación\n4.División");
    printf ("\n\nElige tu opcion: ");
    opc = getchar();
    switch (opc)
    {
        case '1': c = a+b;
            printf ("\nEl resultado es: %d",c);
            break;
        case '2': c = a-b;
            printf ("\nEl resultado es: %d",c);
            break;
        case '3': c = a*b;
            printf ("\nEl resultado es: %d",c);
            break;
        case '4': if (b!=0)
            {
                c = a/b;
                printf ("\nEl resultado es: %d",c);
                d = (float)a/b;
                printf ("\nEl resultado es: %f",d);
            }
            else
                printf ("\nDivisión por cero");
            break;
        default: printf ("\nOpcion invalida");
            break;
    }
}
```

4. Programa que ejemplifica el CAST:

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

```

main()
{
    int a=7, b=3;
    float c;

    c=a/b;
    // La division de dos enteros me da un entero
    printf ("Resultado: %f",c);

    c=(float)a/b; // cast: Forzar a ser de otro tipo
    printf ("\nResultado: %f",c);
    getchar();
}

```

5. Programa que ejemplifica el operador ?:

```

#include <stdio.h>
main()
{
    int a,b;

    printf("Dame a: ");
    scanf("%d",&a);
    printf("Dame b: ");
    scanf("%d",&b);

    a>b ? printf ("\nA es mayor") : printf ("\nB es mayor");
    /* if (a>b)
        printf ("\nA es mayor");
    else
        printf ("\nB es mayor"); */
    getchar();
    getchar();
}

```

6. Programa que obtiene el factorial de un número:

```

#include <stdio.h>
int main()
{
    int n, x=1, i;
    printf ("\nDame el valor de n: ");
    scanf ("%i",&n);
}

```

```

for (i=1; i<=n; i++)
{
    x *= i; /* x = x*i; */
    /* printf ("\n i= %d x= %i",i,x); */
}
/* printf ("\n Fuera del ciclo i= %d x= %i",i,x); */
printf ("\n El factorial es: %i",x);
return 0;
}

```

7. Programa que obtiene el salario de uno o varios trabajadores usando ciclo do – while:

```

#include <stdio.h>
int main()
{
    char otro, nombre;
    int hrstrab;
    float cuotahr, sueldo;
    do
    {
        printf ("\nNombre: ");
        nombre = getchar();
        getchar();
        printf ("\nHoras trabajadas: ");
        scanf ("%d",&hrstrab);
        printf ("\nCuota por hora: ");
        scanf ("%f",&cuotahr);
        sueldo = hrstrab*cuotahr;
        printf ("\nNombre: %c, Sueldo: %f",nombre,sueldo);
        getchar();
        printf ("\nDesea procesar otro empleado S/N? ");
        scanf ("%c",&otro);
        getchar();
    } while (otro=='s' || otro=='S');
    return 0;
}

```

8. Programa que obtiene el salario de uno o varios trabajadores usando ciclo while:

```

#include <stdio.h>
int main()
{
    char otro, nombre;
    int hrstrab;
    float cuotahr, sueldo;
    printf ("\nHay un empleado? ");
    otro = getchar();
    while (otro=='s' || otro=='S')

```

```

{
    getchar();
    printf ("\nNombre: ");
    nombre = getchar();
    getchar();
    printf ("\nHoras trabajadas: ");
    scanf ("%d",&hrstrab);
    printf ("\nCuota por hora: ");
    scanf ("%f",&cuotahr);
    sueldo = hrstrab*cuotahr;
    printf ("\nNombre: %c, Sueldo: %f",nombre,sueldo);
    getchar();
    printf ("\nDesea procesar otro empleado S/N? ");
    scanf ("%c",&otro);
}
return 0;
}

```

9. Programa que obtiene suma de los primeros n – números:

```

#include <stdio.h>
main()
{
    int n, sumatoria=0, k;

    printf ("\nProporciona hasta que numero desea la sumatoria: ");
    scanf ("%d",&n);

    for (k=1; k<=n; k++){
        sumatoria+=k; // sumatoria = sumatoria+k;
        printf ("\nLa sumatoria es: %d",sumatoria);

        getchar();
        getchar();
    }
}

```

10. Programa para el manejo de arreglos:

```

#include <stdio.h>
#define TAM 10
main ()
{
    int arreglo[TAM];
    int i, n, suma=0;

    do {
        printf ("\nDame la dimension del arreglo: ");
        scanf ("%d",&n);
    }
}

```

```

        if (n>TAM)
        {
            printf ("\nError: Se excede el rango");
            printf ("\nProporciona un tamaño menor a %d",TAM);
        }
    } while (n>TAM);
    printf ("\nDa el arreglo: ");
    for (i=0; i<n; i++)
        scanf ("%i",&arreglo[i]);
    for (i=0; i<n; i++)
        suma+=arreglo[i];
    printf ("\nLa suma es: %d\nArreglo:\n",suma);
    for (i=0; i<n; i++)
        printf (" %i",arreglo[i]);

        getchar();
        getchar();
    }

```

11. Programa que invierte una cadena:

```

#include <stdio.h>
#include <string.h>
#define TAM 80
main()
{
    char cad[TAM], c2[TAM];
    int i, j=0, n;

    printf("\nDame la cadena: ");
    scanf ("%^[^\n]s",cad);

    n=strlen(cad);
    printf("\nCadena invertida: ");
    for (i=n-1; i>=0; i--)
    {
        c2[j]=cad[i];
        j++;
        printf("%c",cad[i]);
    }
    c2[j]='\0';
    printf("\nCadena copiada: %s",c2);

    getchar();
    getchar();
}

```

12. Programa que obtiene la suma de matrices:

```
#include <stdio.h>
#define MAX 20
int main ()
{
    int n,m;
    int i,j;
    int A[MAX][MAX], B[MAX][MAX], C[MAX][MAX];
    printf ("\nDame el numero de filas: ");
    scanf ("%i",&n);
    printf ("\nDame el numero de columnas: ");
    scanf ("%d",&m);
    printf ("\nDame la matriz A: ");
    for (i=0; i<n; i++)
    for (j=0; j<m; j++)
    {
        printf ("\nA[%i][%i] =",i+1,j+1);
        scanf ("%d",&A[i][j]);
    }
    printf ("\nDame la matriz B: ");
    for (i=0; i<n; i++)
    for (j=0; j<m; j++)
    {
        printf ("\nB[%i][%i] =",i+1,j+1);
        scanf ("%d",&B[i][j]);
    }
    for (i=0; i<n; i++)
    for (j=0; j<m; j++)
    C[i][j] = A[i][j] + B[i][j];

    printf ("\nLa suma es: ");
    for (i=0; i<n; i++)
    {
        printf ("\n");
        for (j=0; j<m; j++)
        {
            printf ("%i ",C[i][j]);
        }
    }
    getchar();
    return 0;
}
```

13. Programa que maneja estructuras:

```
#include <stdio.h>
#define TAM 20
struct dato
```



```

{
char n_vino[30];
int agno;
} vignedo1[TAM];
/*struct dato vignedo2[TAM]; */
int main()
{
char opc;
int i=0, j, viejo;

do {
printf("1. Introduce vino\n");
printf("2. Mostrar vinos\n");
printf("3. Cosecha mas antigua\n");
printf("4. Salir\n");
printf("\n Elige opcion...");
opc=getchar();
getchar();
switch (opc)
{
case '1': if (i<TAM)
{
printf("\nNombre del vino: ");
scanf("%s",vignedo1[i].n_vino);
printf("\nDa el año: ");
scanf("%d",&vignedo1[i].agno);
i++;
}
else
printf("\n Viñedo lleno \n");
getchar();
break;
case '2': for (j=0; j<i; j++)
{
printf("\nVino: %s",vignedo1[j].n_vino);
printf("\nAño: %i \n",vignedo1[j].agno);
}
break;
case '3': viejo=0;
for (j=1; j<i; j++)
if (vignedo1[viejo].agno>vignedo1[j].agno)
viejo=j;
printf("\nVino: %s \n",vignedo1[viejo].n_vino);
break;
case '4': printf("\n Salir ....\n");
break;
default : printf("\n Error!!! \n");
}
}

```

```

        } while (opc!='4');
        getchar();
        return 0;
    }

```

14. Programa que maneja funciones con paso de parámetros:

```

#include <stdio.h>
void suma(int x,int y)
{
    int c;
    c=x+y;
    x++;
    printf ("\n Resultado = %i",c);
    printf ("\nDirecciones: %i %i",&x,&y);
}
void mult(int *x, int *y)
{
    int c;
    c>(*x)*(*y);
    (*y)++;
    printf ("\n Resultado = %i",c);
    printf ("\nDirecciones: %i %i",x,y);
}
int main()
{
    int a=10, b=11;
    printf ("\nValores: %d %d",a,b);
    printf ("\nDirecciones: %i %i",&a,&b);
    suma (a,b);
    printf ("\nValores: %d %d",a,b);
    mult(&a,&b);
    printf ("\nValores: %d %d",a,b);
    return 0;
}

```

15. Programa que muestra el contenido de un archivo:

```

#include <stdio.h>
#define MAX 80
main()
{
    char nomArch[30], linea[MAX];
    FILE *apF;

    printf("\nDame el nombre del archivo: ");
    gets(nomArch);

    if ((apF = fopen(nomArch,"r"))!=NULL)

```

```

        {
            printf ("\nContenido del archivo: \n");
            while (fgets(linea, MAX, apF)!=NULL)
                printf ("%s",linea);
            fclose(apF);
        }
    else
        printf("\nError en el archivo...");
        getchar();
        getchar();
    }
}

```

16. Programa para el manejo de memoria dinámica:

```

#include <stdio.h>
#include <malloc.h> // Manejo de memoria dinamica
struct info
{
    int dato;
    struct info *sig; // Enlace con el siguiente
    // elemento en la lista ligada
};

struct info *cabeza, *aux;
int menu()
{
    int op;

    printf ("\n1. Insertar \n2. Eliminar ");
    printf ("\n3. Mostrar \n4. Salir \nOpcion: ");
    scanf("%i",&op);

    return op;
}
void insertar(int d)
{
    aux=(struct info *)malloc(sizeof aux);
    if (aux != NULL)
    {
        aux->dato=d;
        if (cabeza ==NULL)
        {
            aux->sig=NULL;
            cabeza=aux;
        }
    }
    else
    {
        aux->sig=cabeza;
        cabeza=aux;
    }
}

```

```

    }
}
else
printf("\nMemoria insuficiente");
}
void eliminar()
{
if (cabeza!=NULL)
{
aux=cabeza;
cabeza=aux->sig;
printf("\nDato a eliminar: %d",aux->dato);
free(aux);
}
else
printf("\nLista vacia");
}
void mostrar()
{
aux=cabeza;
while(aux!=NULL)
{
printf("\n %d", aux->dato);
aux=aux->sig;
}
}
main ()
{
int opc, d;

cabeza=NULL;
do {
opc =menu();
getchar();
switch(opc)
{
case 1: printf("\nDa el dato: ");
scanf("%d",&d);
insertar(d);
break;
case 2: eliminar();
break;
case 3: mostrar();
break;
case 4: printf("\nSaliendo...");
break;
default: printf("\nOPcion invalida");
break;
}
}
}

```

```

    }
} while (opc!=4);
}

```

17. Programa que guarda a archivo estructuras:

```

#include <stdio.h>
#include <string.h>
#define MAX 5
struct agenda
{
    char nombre[30];
    int tel;
    char dir[40];
    int band;
};
char menu()
{
    printf ("\n1. Altas \n2. Consulta \n3. Buscar");
    printf ("\n4. Bajas \n5. Salir \nOpcion: ");
    return getchar();
}
void alta(struct agenda *amigos)
{
    int i=0;
    while (i<MAX && amigos[i].band!=0)
        i++;
    if (i<MAX)
    {
        printf ("\nDame el nombre: ");
        gets(amigos[i].nombre);
        printf ("\nDame la direccion: ");
        gets (amigos[i].dir);
        printf ("\nDame el telefono: ");
        scanf ("%d",&amigos[i].tel);
        amigos[i].band=1;
        getchar();
    }
    else
        printf ("\nEspacio insuficiente");
}
void consulta(struct agenda *amigos)
{
    int j;

    printf("\nDatos en la agenda:");
    for (j=0; j<MAX; j++)
        if (amigos[j].band!=0)
            {

```

```

        //printf ("\nBandera: %i",amigos[j].band);
        printf ("\nNombre: %s",amigos[j].nombre);
        printf ("\nDireccion: %s",amigos[j].dir);
        printf ("\nTelefono: %i\n",amigos[j].tel);
        getchar();
    }
}

void buscar(struct agenda *ami)
{
    int i=0;
    char nombre[30];

    printf("\nDame el nombre a buscar: ");
    gets(nombre);
    while (i<MAX && strcmp(ami[i].nombre,nombre)!=0)
        i++;
    if (i==MAX)
        printf ("\nDato no encontrado");
    else
        if (ami[i].band==1)
            {
                printf ("\nNombre: %c",ami[i].nombre);
                printf ("\nDireccion: %c",ami[i].dir);
                printf ("\nTelefono: %i\n",ami[i].tel);
            }
        else
            printf ("\nDato no encontrado");
    }

void baja(struct agenda *ami)
{
    int i=0;
    char nombre[30], op;

    printf("\nDame el nombre a dar de baja: ");
    gets(nombre);
    while (i<MAX && strcmp(ami[i].nombre,nombre)!=0)
        i++;
    if (i==MAX)
        printf ("\nDato no encontrado");
    else
        if (ami[i].band==1)
            {
                printf ("\nNombre: %s",ami[i].nombre);
                printf ("\nDireccion: %s",ami[i].dir);
                printf ("\nTelefono: %i\n",ami[i].tel);
                printf ("\nSeguro quieres darlo de baja? ");
                op = getchar();
                if (op=='s' || op=='S')

```

```

    {
        printf ("\nDando de baja...");
        ami[i].band=0;
    }
    getchar();
}
else
printf ("\nDato no encontrado");
}
main()
{
    struct agenda amigos[MAX];
    int i, j;
    char opc;
    FILE *apF;

    for (i=0; i<MAX; i++)
        amigos[i].band = 0;
        // Se abre de solo lectura para obtener los datos
    if ((apF=fopen("datos.dat","r"))==NULL)
    {
        apF=fopen("datos.dat","w"); // Si no existe lo crea
    }
    else
    {
        //printf ("\nTamaño: %d", sizeof(struct agenda));
        for (j=0; j<MAX; j++)
            fread (&amigos[j],sizeof(struct agenda),1,apF);
        fclose(apF); // Se cierra el archivo para "reiniciar" el apuntador
    }
    do
    {
        opc = menu();
        getchar();
        switch (opc)
        {
            case '1': alta(amigos);
                break;
            case '2': consulta(amigos);
                break;
            case '3': buscar(amigos);
                break;
            case '4': baja(amigos);
                break;
            case '5': printf ("\nSaliendo...");
                if ((apF=fopen("datos.dat","w"))!=NULL)
                { // Se guardan todos los datos
                    for (j=0; j<MAX; j++)

```

```

    {
    //printf ("\nBandera: %d",amigos[j].band);
    //if (j%23==0)
    // getchar();
    fwrite(&amigos[j],sizeof(struct agenda),1,apF);
    }
fclose(apF);
}
else
    printf("\nError en archivo");
    break;
    default: printf ("\nOpcion invalida...");
    break;
    }
} while (opc!='5');
}

```

18. Programa que Que gestiona las notas de una clase de 20 alumnos de los cuales sabemos el nombre y la nota. El programa debe ser capaz de:

- Buscar un alumno.
- Modificar su nota.
- Realizar la media de todas las notas.
- Realizar la media de las notas menores de 5.
- Mostrar el alumno que mejores notas ha sacado.
- Mostrar el alumno que peores notas ha sacado.

```

#include <stdio.h>
#include <stdlib.h>

struct alumno {
    char nombre[50];
    float nota;
};

int main(int argc, char *argv[])
{
    struct alumno alum,alumnos[5];

    int x,opcion=1;
    float sum=0,cont=0,mejor,peor;

```



```

for (x=0;x<5;x++)
{
    printf("Introduzca nombre alumno:");
    gets(alumnos[x].nombre);
    gets(alumnos[x].nombre);
    printf("Introduzca nota:");
    scanf("%f",&alumnos[x].nota);
}

while ((opcion==1 || opcion==2 ||
        opcion==3 || opcion==4 ||
        opcion==5 || opcion==6) && (opcion!=7))
{

    printf("1- Buscar un alumno\n");
    printf("2- Modificar nota\n");
    printf("3- Media de todas las notas\n");
    printf("4- Media de todas las notas inferiores a 5\n");
    printf("5- Alumno con mejores notas\n");
    printf("6- Alumno con peores notas\n");
    printf("7- Salir\n");
    printf("Introduzca una opción: ");
    scanf("%d",&opcion);

    if (opcion==1)
    {
        printf("Introduzca un nombre: ");
        gets(alum.nombre);
        gets(alum.nombre);

        for(x = 0; x < 5;x++)
        {
            if (strcmp(alumnos[x].nombre,alum.nombre)==0)
            {
                printf("\nNombre: %s\n",alumnos[x].nombre);
                printf("Nota: %f\n",alumnos[x].nota);
            }
        }
        printf("\n\n");
    }
    else if (opcion==2)
    {
        printf("Introduzca un nombre: ");
        gets(alum.nombre);
    }
}

```

```

gets(alum.nombre);

for(x = 0; x < 5;x++)
{
    if (strcmp(alumnos[x].nombre,alum.nombre)==0)
    {
        printf("Introduzca una nota: ");
        scanf("%f",&alumnos[x].nota);
        printf("\nNota modificada.");
    }
}
printf("\n\n");
else if (opcion==3)
{
    sum=0;
    for(x = 0; x < 5;x++)
    {
        sum=sum+alumnos[x].nota;
    }
    printf("\nLa media de las notas es de: %f \n",(sum/5));
}
else if (opcion==4)
{
    sum=0;
    cont=0;
    for(x = 0; x < 5;x++)
    {
        if (alumnos[x].nota<5)
        {
            sum=sum+alumnos[x].nota;
            cont++;
        }
    }
    printf("\nLa media de las notas inferiores a 5 es: %f \n",sum/cont);
}
else if (opcion==5)
{
    mejor=0;
    for(x = 0; x < 5;x++)
    {
        if (alumnos[x].nota>mejor)
        {
            mejor=alumnos[x].nota;

```

```

        alum.nota=alumnos[x].nota;
        strcpy(alum.nombre,alumnos[x].nombre);
    }
}
printf("\nEl alumno con mejores notas es: %s \n",alum.nombre);
}
else if (opcion==6)
{
    peor=10;
    for(x = 0; x < 5;x++)
    {
        if (alumnos[x].nota<peor)
        {
            peor=alumnos[x].nota;
            alum.nota=alumnos[x].nota;
            strcpy(alum.nombre,alumnos[x].nombre);
        }
    }
    printf("\nEl alumno con peores notas es: %s \n",alum.nombre);
}
}

system("PAUSE");
return 0;
}

```