

Practica No 8

Objetivo general:

En esta práctica realizamos programas usando funciones y punteros para lograr aprender su funcionamiento.

Desarrollo:

En esta ocasión realizamos dos programas el primero de ellos consta de convertir las palabras a un código llamado Morse por lo que a manera de metodología usamos el código Morse como base al ver que este se basa solamente de líneas y puntos vimos que para sacar las señales por leds y por sonido necesitaríamos crear una función una de ellas seria para el punto y otra para la línea. Y depuse usando estas funciones estructuraríamos otras funciones que serian de letras después evaluamos si en la cadena existe esa letra mandamos a llamar a su función.

Por lo que el programa quedo de la siguiente forma:

```
/*
FECHA: 14-NOV-06
PROGRAMA: 26
OBJETIVO: /*COMENZAR A UTILIZAR LAS FUNCIONES DENTRO DE NUESTROS PROGRAMAS
PARA ASI VER SU GRAN AYUDA DENTRO DE PROGRAMAS QUE SON MUY REPETITIVOS Y
OBSERVAR QUE ESTOS PROGRAMAS REDUCEN DE GRAN FORMA NUESTRO CODIGO
FUENTE*/
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <string.h>
void mar ()
{
int x;
textbackground (RED);
textcolor (GREEN);
clrscr ();
for (x=1;x<=80;x++)
{
gotoxy (x,1);
printf ("*");
delay (25);
}
for (x=1;x<=49;x++)
{
gotoxy(80,x);
printf ("*");
delay (25);
}
for (x=80; x>=1;--x)
{
gotoxy (x,49);
printf ("*");
}
```

```

delay (25);
}
for (x=49;x>=1;--x)
{
gotoxy (1,x);
printf ("*");
delay (25);
}
}
void punto()
{
printf (".");
outport (0x378,1);
sound (200);
delay (100);
outport (0x378,0);
nosound ();
}
void linea ()
{
printf ("-");
outport (0x37,3);
sound (200);
delay (200);
outport (0x3789,0);
nosound ();
}

void a ()
{
punto ();
linea ();
printf (" ");
}

void b ()
{
linea ();
punto ();
punto ();
punto ();
printf (" ");
}

void c ()
{
linea ();
punto ();
linea ();
punto ();
printf (" ");
}

void d()
{
linea ();
}

```

```
punto ();
punto ();
printf (" ");
}
void e()
{
punto ();
printf (" ");
}
void f ()
{
punto ();
punto ();
punto ();
linea ();
punto ();
printf (" ");
}
```

```
void g ()
{
linea ();
linea ();
punto ();
printf (" ");
}
```

```
void h()
{
punto ();
punto ();
punto ();
punto ();
printf (" ");
}
```

```
void i ()
{
punto ();
punto ();
printf (" ");
}
```

```
void j()
{
punto ();
linea ();
linea ();
linea ();
printf (" ");
}
```

```
void k ()
{
linea ();
punto ();
}
```

```
punto ();  
printf (" ");  
}
```

```
void l()  
{  
punto ();  
linea ();  
punto ();  
punto ();  
printf (" ");  
}
```

```
void m()  
{  
linea ();  
linea ();  
printf (" ");  
}
```

```
void n ()  
{  
linea ();  
punto ();  
printf (" ");  
}
```

```
void o()  
{  
linea ();  
linea ();  
linea ();  
printf (" ");  
}
```

```
void p ()  
{  
punto ();  
linea ();  
linea ();  
punto ();  
printf (" ");  
}
```

```
void q ()  
{  
linea ();  
linea ();  
punto ();  
linea ();  
printf (" ");  
}
```

```
void r ()  
{  
punto ();  
linea ();  
}
```

```
punto ();  
printf (" ");  
}
```

```
void s()  
{  
punto ();  
punto ();  
punto ();  
printf (" ");  
}
```

```
void t ()  
{  
linea ();  
printf (" ");  
}
```

```
void u ()  
{  
punto ();  
punto ();  
linea ();  
printf (" ");  
}
```

```
void v ()  
{  
punto ();  
punto ();  
punto ();  
linea ();  
printf (" ");  
}
```

```
void w()  
{  
punto ();  
linea ();  
linea ();  
printf (" ");  
}
```

```
void x ()  
{  
linea ();  
punto ();  
punto ();  
linea ();  
printf (" ");  
}
```

```
void y ()  
{  
linea ();  
punto ();  
}
```

```
linea ();
linea ();
printf (" ");
}
```

```
void z ()
{
linea ();
linea ();
punto ();
punto ();
printf (" ");
}
```

```
void main ()
{
int x1;
int ptrx1;
char ptral;
char a1[100];
clrscr ();
ptrx=&x;
ptral=&a;
mar ();
printf ("\n*\tintroduce la palabra a transformar\n");
gotoxy (5,5);
gets (ptral);
printf ("*\t");
for (*ptrx1=0;*ptrx1<strlen (a1);*ptrx1=*ptrx1+1)
{
printf ("%c\t", a1[x1]);
if (ptral[*ptrx1]=='a')
a ();
if (ptral[*ptrx1]=='b')
b ();
if (ptral[*ptrx1]=='c')
c ();
if (ptral[*ptrx1]=='d')
d ();
if (ptral[*ptrx1]=='e')
e ();
if (ptral[*ptrx1]=='f')
f ();
if (ptral[*ptrx1]=='g')
g ();
if (ptral[*ptrx1]=='h')
h ();
if (ptral[*ptrx1]=='i')
i ();
if (ptral[*ptrx1]=='j')
j ();
if (ptral[*ptrx1]=='k')
k ();
if (ptral[*ptrx1]=='l')
l ();
if (ptral[*ptrx1]=='m')
```

```

    m ();
    if (ptr1[*ptrx1]=='n')
        n ();
    if (ptr1[*ptrx1]=='o')
        o ();
    if (ptr1[*ptrx1]=='p')
        p ();
    if (ptr1[*ptrx1]=='q')
        q ();
    if (ptr1[*ptrx1]=='r')
        r ();
    if (ptr1[*ptrx1]=='s')
        s ();
    if (ptr1[*ptrx1]=='t')
        t ();
    if (ptr1[*ptrx1]=='u')
        u ();
    if (ptr1[*ptrx1]=='v')
        v ();
    if (ptr1[*ptrx1]=='w')
        w ();
    if (ptr1[*ptrx1]=='x')
        x ();
    if (*ptr1[*ptrx1]=='y')
        y ();
    if (ptr1[*ptrx1]=='z')
        z ();
    if (ptr1[*ptrx1]==32)
        printf ("\n*\t");
    }
getch ();
}

```

/*CONCLUSIONES: EN ESTE PROGRAMA VEMOS LA GRAN IMPORTANCIA DE NUESTRO PROGRAMA YA QUE EN ESTE PROGRAMA QUE ES MUY REPETITIVO PODEMOS HACER QUE SEA MAS CORTO NUESTRO CODIGO FUENTE*/

LA corrida de nuestro programa es el siguiente:


```

void mar ()
{
int x;
textbackground (RED);
textcolor (GREEN);
clrscr ();
for (x=1;x<=80;x++)
{
gotoxy (x,1);
printf ("*");
delay (25);
}
for (x=1;x<=49;x++)
{
gotoxy(80,x);
printf ("*");
delay (25);
}
for (x=80; x>=1;--x)
{
gotoxy (x,49);
printf ("*");
delay (25);
}
for (x=49;x>=1;--x)
{
gotoxy (1,x);
printf ("*");
delay (25);
}
}
void main ()
{
int x,y,u,punto,con=0;
char a[30],b[30],d[30];
clrscr ();
mar ();
for (u=0;u<30;u++)
{
a[u]='\x0';
b[u]='\x0';
d[u]='\x0';
}
flushall ();
printf ("Introduce la cifra\n\n");
gets (a);
for (x=0;x<19;x++)
{
if (a[x]=='.')
punto=x;
}
for (x=0,y=punto+1;x<2;x++,y++)
{
b[2]='\x0';
b[x]=a[y];
}
}

```

```

for (x=0;x<19;x++)
{
if (a[x]=='.')
x=20;
else
d[x]=a[x];
d[x+1]=NULL;
}
y =strlen (d);
if (y>8)
{
for (x=0;x<y-8;x++)
{
d[x]='@';
}
}
for (x=0;x<y;x++)
{
if (d[x]!='@')
{
a[con]=d[x];
a[con+1]=NULL;
con++;
}
}
printf ("La validacion del cheque es por:\n\n");
for (x=1;x<=8-y;x++)
{
printf ("0");
}
printf ("%s.%s",a,b);
getch ();
}
/*CONCLUSIONES: EN ESTA PRACTICA OBSERVAMOS QUE LAS FUNCIONES NO CAMBIAN EN
NADA LA ESTRUCTURA DE LOS CICLOS DE REPETICIÓN ADEMÁS DE QUE SE PUEDEN
ANIDAR LAS FUNCIONES TAMBIÉN*/

```

La corrida del programa se ve de la siguiente forma:

