Desarrolle los siguientes puntos.

1. **(20%)** Se dice que una matriz tiene un "punto de silla" si alguna posición de la matriz es el menor valor de su fila, y a la vez el mayor de su columna. Realice una función que determine si una matriz tiene algún "punto de silla".

```
int menorfil(int mat[][MAXCOL], int fila, int c)
{
   int ind;
   int menor = mat[fila][0];

   for (ind = 1; ind < c; ind ++ )
      if ( menor > mat[fila][ind] )
         menor = mat[fila][ind];
   return menor;
}

int mayorcol(int mat[][MAXCOL], int f, int columna)
{
   int ind;
   int mayor = mat[0][columna];

   for (ind = 1; ind < f; ind ++ )
      if ( mayor < mat[ind][columna] )
         mayor = mat[ind][columna];
   return mayor;
}


int punto_silla(int mat[][MAXCOL], int f, int c)
{
   int ind1, ind2;

   for ( ind1 = 0; ind1 < f; ind1 ++ )
      for ( ind2 = 0; ind2 < f; ind2 ++ )
      {
         if ( mat[ind1][ind2] == menorfil(mat,ind1,c) &&
               mat[ind1][ind2] == mayorcol(mat,f,ind2) )
            return 1;

      }

   return 0;
}
```

2. **(30%)** Dada la cadena de caracteres s, se desea saber cuál es el caracter que más se repite de dicha cadena y cuantas veces se repite. Realice la función modachar utilizando el prototipo que usted considere mejor.

```
int modachar(char s[],char *moda)
{
   int ind, repite,repitemod = veces(s,s[0]);

   *moda = s[0];

   for ( ind = 1; s[ind];  ind ++ )
   {
       repite = veces(s,s[ind]);
       if ( repite > repitemod )
       {
          *moda = s[ind];
          repitemod = repite;
       }
   }

   return repitemod;
}

// Cuantas veces está 'c' en 's'
int veces(char s[], char c)
{
   int ind, contador = 0;

   for ( ind = 0; s[ind];  ind ++ )
      if ( c == s[ind] )
         contador ++;

   return contador;
}
```

3. **(30%)** Realice la función **void wordcap(char *s)** la cual formatea el string **s** poniendo la primera letra de cada palabra contenida en **s** en mayúscula y el resto en minúscula. Es decir, si **s = "la casa blanca Del pantano"**, luego de la llamada a la función **wordcap(s)**, **s = "La Casa Blanca Del Pantano"**.

```
void wordcap(char *s)
{
   int ind;

   for ( ind = 0; *(s+ind);  ind ++ )
   {
      if ( *(s+ind) >= 'A' && *(s+ind) <= 'Z' )
         *(s+ind) += 32;

      if (*(s+ind) != ' ' )
      {
         if ( ind == 0 )
         {
            if (*(s+ind) >= 'a' && *(s+ind) <= 'z' )
               *(s+ind)-= 32;
         }
         else
         {
            if (*(s+ind-1) == ' ')
               if (*(s+ind) >= 'a' && *(s+ind) <= 'z' )
                  *(s+ind) -= 32;
         }
      }
   }
   return;
}
```

4. **(20%)** Un estudiante posee los siguientes atributos: matrícula de 8 caracteres, nombre1 de 30 caracteres, nombre2 de 35 caracteres, apellido1 de 25 caracteres, apellido2 de 25 caracteres, carrera de 4 caracteres, categoría de pago de 3 caracteres, nacionalidad de 3 caracteres, dirección de 256 caracteres. Defina un tipo de datos que pueda almacenar un estudiante con estos atributos, realice además una función que permita capturar un estudiante.

```
typedef struct{
   char matricula[8], nombre1[30], nombre2[35],
        apellido1[25], apellido2[25], carrera[4],
        categoria_pago[3], nacionalidad[3],
        direccion[256];
}EST;
```

```
// Primera versión

void cap_estud(EST *est)
{
   printf("Matricula: ");gets(est->matricula);
   printf("Nombre1: ");gets(est->nombre1);
   printf("Nombre2: ");gets(est->nombre2);
   printf("Apellido1: ");gets(est->apellido1);
   printf("Apellido2: ");gets(est->apellido2);
   printf("Carrera: ");gets(est->carrera);
   printf("Categoria de pago: ");gets(est->categoria_pago);
   printf("Nacionalidad: ");gets(est->nacionalidad);
   printf("Direcci%cn: ",162);gets(est->direccion);
   return;
}

// Segunda Versión

EST cap_estud2(void)
{
   EST est;
   printf("Matricula: ");gets(est.matricula);
   printf("Nombre1: ");gets(est.nombre1);
   printf("Nombre2: ");gets(est.nombre2);
   printf("Apellido1: ");gets(est.apellido1);
   printf("Apellido2: ");gets(est.apellido2);
   printf("Carrera: ");gets(est.carrera);
   printf("Categoria de pago: ");gets(est.categoria_pago);
   printf("Nacionalidad: ");gets(est.nacionalidad);
   printf("Direcci%cn: ",162);gets(est.direccion);
   return est;
}
```