Pontificia Universidad Católica Madre y Maestra **Valor (25%)**
Facultad de Ciencias de La Ingeniería **Segundo Parcial**
Departamento de Ingeniería de Sistemas y Computación
Algoritmos Fundamentales-ISC-201 Matrícula:_____

1. Se tienen la siguiente declaración:

```
typedef struct
{
    char matricula[8], nombre[20], carrera[3];
    float puntos;
    int creditos;
} EST;
```

Realice una función **int existe_estudiante(EST *data, char *matricula, int n),** retorne 1 si el estudiante con la matrícula contenida en **"matricula"** está en los datos accesados desde el puntero **"data"** hasta **"n"** estudiantes. **(25%)**

```
int existe_estudiante(EST *data, char *matricula, int n)
{
   int ind;

   for ( ind = 0; ind < n; ind ++ )
      if (strcmp((data+ind)->matricula,matricula )== 0 )
         return 1;

   return 0;
}
```

Realice una función que retorne el estudiante con el índice mayor. Defina esta función como lo crea más conveniente. **(25%)**

```
EST estudindmayor(EST *data, int n)
{
   int ind, pos_mayor = 0;
   float indicemayor = data->puntos / data->creditos;

   for ( ind = 1; ind < n; ind ++ )
      if ( indicemayor < (data+ind)->puntos / (data+ind)->creditos
)
      {
         pos_mayor = ind;
         indicemayor = (data+ind)->puntos / (data+ind)->creditos;
      }

   return *(data+posmayor); // data[posmayor];
}
```

Profesor: Alejandro Liz **11 de octubre de 2008**

2. **(20%)** Un string es palíndrome si se lee igual al derecho y al revés, por ejemplo "anitalavalatina" al revés es "anitalavalatina", "dabalearrozalazorraelabad" al revés es "dabalearrozalazorraelabad", "ana" al revés es "ana". Los anteriores son todos palíndrome. Si se tiene la función **void invetir(char s[ ]),** la cual invierte el string **s**. Realice la función **int espalindrome(char s[ ])**, la cual retorna **1** si el string **s** es palíndrome y **cero** si no lo es.

```
int espalindrome(char s[ ])
{
   char *cs;
   int palindrome = 0;

   cs = (char *) malloc(strlen(s)+1);

   strcpy (cs,s);

   invertir(cs);

   if ( strcmp(cs,s) == 0 )
      palindrome = 1;

   free(cs);
   return palindrome;
}
```

3. **(30%)** 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;
}
```

4. **(10% OPCIONAL)** El valor de π está determinado por una serie como sigue: π = 4 * (1 – 1/3 + 1/5 – 1/7 + 1/9...). Haga una función recursiva que retorne el valor de π evaluando n términos de la serie.

```
float pi_r(int n)
{
   if ( n == 1 )
      return 4;

   return 4 * (float) 1 / (2 * n - 1) * (n % 2 ? 1 : -1);
}
```